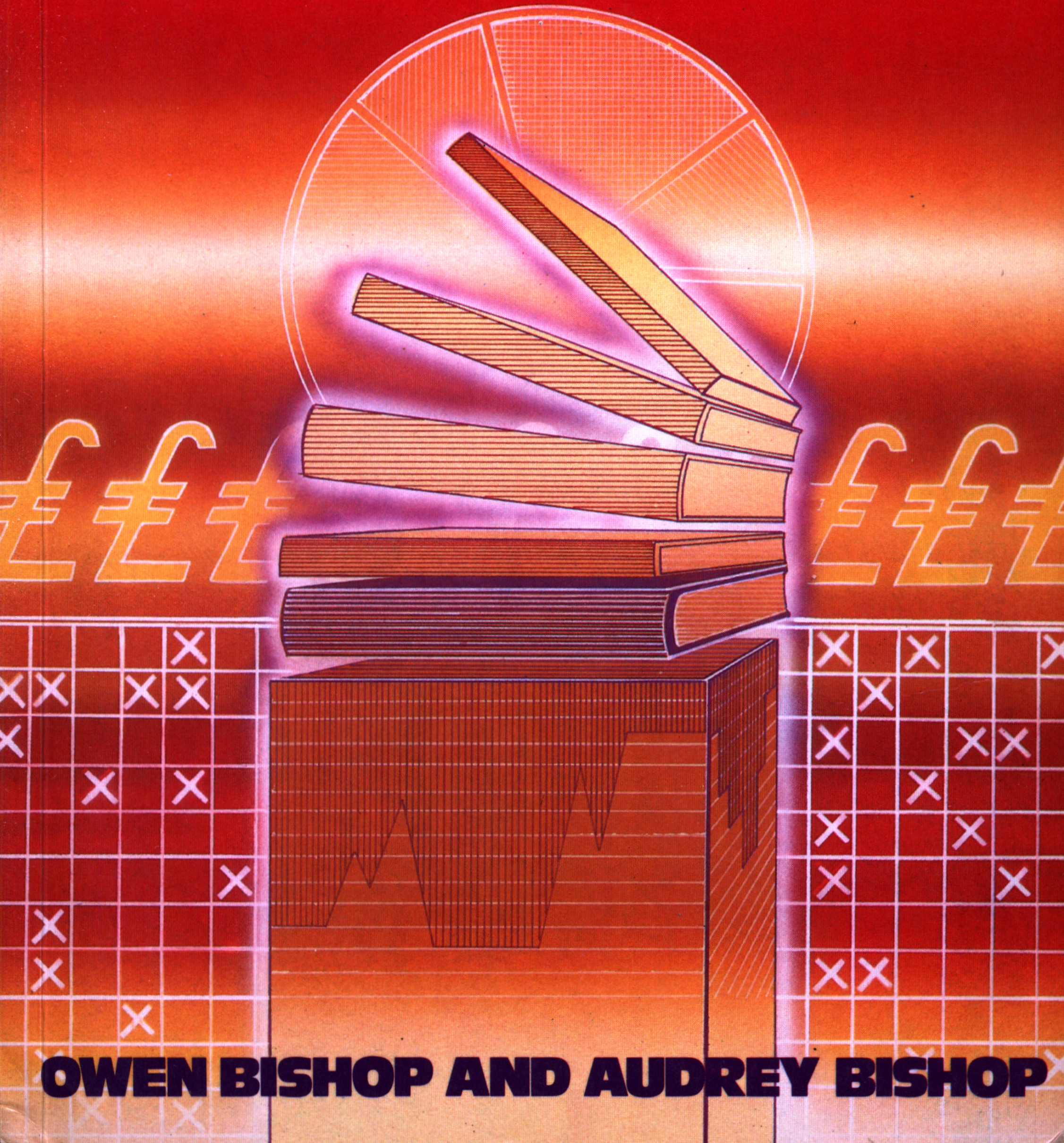
PRACTICAL PROGRAMS FOR THE AMSTRAD CPC464



OWEN BISHOP AND AUDREY BISHOP

Practical Programs for the Amstrad CPC464

Other books for Amstrad users

Sensational Games for the Amstrad CPC464
Jim Gregory
0 00 383121 3

40 Educational Games for the Amstrad CPC464
Vince Apps
0 00 383119 1

Adventure Games for the Amstrad CPC464
A. J. Bradbury
0 00 383078 0

Introducing Amstrad Machine Code
Ian Sinclair
0 00 383079 0

Filing Systems and Databases for the Amstrad CPC464
A. P. Stephenson and D. J. Stephenson
0 0 383102 7

Amstrad Computing
Ian Sinclair
0 00 383120 5

Practical Programs for the Amstrad CPC464

Owen Bishop and Audrey Bishop

COLLINS
8 Grafton Street, London W1

Collins Professional and Technical Books
William Collins Sons & Co. Ltd
8 Grafton Street, London W1X 3LA

First published in Great Britain by
Collins Professional and Technical Books 1985

Distributed in the United States of America
by Sheridan House, Inc.

Copyright © Owen Bishop and Audrey Bishop 1985

ISBN 0 00 383082 9

Typeset by V & M Graphics Ltd, Aylesbury, Bucks
Printed and bound in Great Britain by
Mackays of Chatham, Kent

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Contents

Acknowledgement vi

Key Points vii

1 Cashflow 1

2 Gottit! 17

3 Stock Take 23

4 Pie Chart 38

5 Index 45

6 Compare It 55

7 Space Plan 63

8 Accounts 83

9 Phone Call Coster 105

10 Info File 116

11 Pools Punter 132

12 Trend Finder 146

13 Supersound 151

14 What Next? 163

Acknowledgement

We would like to thank Amsoft Ltd
for helpful technical advice.

Key Points

To obtain error-free listings, all programs were listed on a printer attached to the micro immediately after testing the final version. All the programs should work perfectly, provided that they are correctly keyed in. The description of each program has a section which deals with particular points about keying in the program. Below are given some general points that apply to all the programs.

A mistake which is easy to make is to type ‘O’ (letter ‘oh’) instead of ‘0’ (figure ‘zero’). The letter (‘oh’) is used in BASIC keywords such as ‘FOR’ and ‘OPENOUT’, but is never used alone in this book as a variable name. In a few instances, it is part of a variable name, e.g. MO$, used to represent a month number. In the listing, and on the screen, the ‘zero’ has a ‘slash’ through it.

Another possible confusion is between ‘I’ (letter ‘eye’), ‘l’ (small letter ‘el’), ‘1’ (figure ‘one’) and ¦ (two vertical lines, one above the other). They are usually easily distinguishable from each other in the listings and we have tried to point out all instances in which confusion is likely to arise. Read the *Keying in* section of each chapter for details.

The semicolon (;) is an essential item in many PRINT statements. Watch out for this, especially at the ends of program lines. It may be more difficult to spot in a multistatement line such as:

```
FOR J=1 TO 10:PRINT M$(J);:NEXT
```

If you leave out the semicolon, the display on the screen will be spoiled.

It is essential to note the difference between:

“”   and   “ ”

The example on the left has nothing between the two quotes. In other words, it is an empty string, usually referred to as a null string. The example on the right has a space between the double quotes and is therefore not an empty string. A space is a character, just as much as any other character, such as 'A' or '9'. To leave out a space when it should be there, or to type a space when there should be no space, can cause a program to crash.

There is one typing error that is easy to make, but which is extremely difficult to spot. It is most likely to occur when a program line ends exactly on the right-hand edge of the screen. The error is to forget to key ENTER before starting to type in the next line. The effect of this omission is that the next line becomes tagged on to the line which has just been typed. The computer will not accept a line which is tagged on to the end of another one, so it naturally reports an error when the program is run. The type of error it reports may vary, depending on what statements were in the lines concerned. Unfortunately, there is no special error message to say 'You forgot to type ENTER', so the error message you do get can be misleading. Even more confusing is the fact that when you list that part of the program, the listing on the screen looks perfectly correct. The first of the two lines ends at the right-hand edge of the screen; the second, tagged-on, line is displayed as expected, beginning at the left-hand edge of the screen. There is nothing to indicate that the two are joined together. If the computer reports an error at a particular line and you can see nothing wrong with your typing, list that line by itself. For example, if the message 'Syntax error in 410' appears, list line 410 on its own. You might then see this displayed on the screen:

```
LIST 410
410 FOR J=1 TO 250:D$(J,2)=E$(J+X):NEXT
420 LOCATE #2,10,5:PRINT K$
```

Line 420 has appeared too! If this happens, retype both lines, remembering to press ENTER at the end of each line.

1

Cashflow

This is a spreadsheet program, specially designed to help you plan your family finances. It will help you forecast your spending for a year or slightly more. You tell the computer how you want it to work out your monthly income or your monthly expenditure on each item of your home budget, and it automatically does this for the whole planning period. This is helpful in itself, but the program really shows its merits when you decide to alter the way one or more of the items is calculated. You simply type in the alteration and the computer recalculates that item and any other items that are related to it. The example which follows later explains in more detail what CASHFLOW can do.

Being able to plan accurately, you are more easily able to foresee periods when you are likely to be short of cash. It is better to know about this in advance, so that you can cut down expenses in time, or can arrange for extra income or a loan. On the other hand, if there is something you want to buy now and CASHFLOW shows that you will have enough spare cash in the foreseeable future, you can confidently make your purchase immediately.

But please remember that the computer is basing its calculations on information that *you* feed into it. If you give it the wrong information, or do not clearly tell it what to do with that information, the results it hands back to you will be wrong. Treat the program as an aid to intelligent money management.

Using the program

The first thing to decide is the period for which you wish to work out your cashflow. This can be for any number of months from 2 to 14. A 14-month period allows you to plan for a full year ahead and also to see how your finances are likely to be as the next year begins. The first question the program asks is 'Starting from which month?'. Key in a number between 1 and 12 in reply to this question. The number is the number of the month *in the calendar year*; 1 = January, 2 = February, and so on. Next you are asked how many months you wish the cashflow to cover. Key in any number from 2 to 14.

Finally, you are asked 'How many items?'. This is the number of separate items of income or expenditure that you wish to include in the cashflow. If you are in doubt, it is better to ask for more items than you think you will need. It is all too easy to remember an essential item when you are halfway through the program, but at that stage it is not possible to increase the number of items. On the other hand, if you have a spare item or two that is subsequently not used, it makes no difference to the calculations.

The screen clears and then the column headings of the spreadsheet appear. These look like the top two rows of Fig. 1.1. To set up this example, the starting month selected was '4' (April), and the number of months chosen was 14. The cashflow runs from April of one year until May of the next year. Note that the columns are numbered from 1 to 14. The first April is column 1, and the second May is column 14. These are the numbers used to identify each month from now on.

At this stage, there is a letter of the alphabet at the left end of each row of the table. There is one lettered row for each item you have chosen to work with.

Further input to the program is at the bottom left of screen. The first question you will see there is 'Item A Name:'. Type in the name of the first item, noting that only the first seven letters of the name will be used to label the spreadsheet. In the example of Fig. 1.1, the first name typed in was 'INCOME'. As soon as ENTER is pressed, the name appears on row A. The next question is 'Initial value (0-9999)'. On most occasions you key in zero, but the example given later will explain in more detail what you should do at this stage.

Now the computer wants to know how it is to calculate the values in the spreadsheet for each month of the period. You first see the message 'Period 1–'. You may wish the computer to perform the same calculation for all months of the period. If so, type '14' (or whatever number of months your cashflow is covering). You may wish the calculation to cover only, say, the first four months of the period. In this case, type 4. If you want the calculation to be for the first month only, type 1.

As soon as you have typed a number and pressed ENTER, an '=' appears after it. This is the computer's way of inviting you to tell

```
            Apr  May  Jun  Jul  Aug  Seo  Oct  Nov  Dec  Jan  Feb  Mar  Apr  May
            1    2    3    4    5    6    7    8    9    10   11   12   13   14
A INCOME     550  550  550  550  550  550  550  550  550  600  600  600  600  600
B HOUSE      150  100  100  100  100  100  150  100  100  100  100  100  150  100
C CAR         65   65   65   95   65   65   65   65   65   65   20   20   20   20
D FOOD       180  180  180  180  180  180  180  180  180  180  180  180  180  180
E FUEL        15   15   15   15   15   15   30   30   30   30   30   30   15   15
F HOLIDAY    108  132  156  176    0   24   40   62   84  112  146  180  209  245
G SAVINGS    367  460  534  567  620  662  639  664  684  744  831  901  927  991
H CLOTHES     73   92  107  113  124  132  128  133  137  149  166  180  185  198
```

Fig. 1.1. Typical household cashflow.

it what calculation is to be done for the month or months you have specified. 'Doing a calculation' sounds impressive, but often only a very simple expression is to be typed in here. If this is your first run through, type in a number (e.g. 50). Otherwise see the example given later.

If you have specified a period up to the last month of the cashflow, you will then be asked to repeat the above entries for item B and subsequent items in order. If you have specified, say, the first four months as your first period, the next message to appear will be 'Period 5–'. Now you can select a second period, in which the basis for calculation can be entirely different. As before, this period can run to the end of the cashflow (type the number of the last month, e.g. 14), or for a few months, or for only one month. The computer will continue asking you to key in details until you have told it what to do for every month of the cashflow. Before proceeding to the next item it always asks 'ALL OK?'. At this stage, check your entries and if there are mistakes, key 'N'. The area at the bottom of the screen clears. The item name in the table clears also. You can then start again to re-enter item name, initial value, and the calculations for as many periods as you want. When all is correct, key 'Y' to go to the next item.

When all items have been keyed in, you will see columns of figures appearing on the spreadsheet. At the end, it will look something like Fig. 1.1. This is your cashflow chart, which you can now examine at leisure. It may have taken you some time to key in the details for each item, but most of the work is done now. From now on, making changes to one row at a time takes little effort, yet can produce the most interesting results. At the bottom of the screen you will see the message 'Amend item? (Key letter)'. If you want to alter row C, for example, press key 'C'. You will then be able to alter any detail for that row, except its name. The procedure is just the same as that for entering the details initially, as already described.

As soon as the amended details have been entered, all figures in the cashflow are recalculated and displayed. You will be able to spot the new values replacing old ones where necessary. Once again you are asked 'Amend item?'. There is no end to the program; you can carry on amending items indefinitely and each time you will see the latest version of your cashflow chart. If you have a printer attached to your computer and wish for a printout of the cashflow chart, enter 'P' in response to 'Amend item?'. The cashflow chart will then be printed out, just as it appears on the screen. After this, you are returned to 'Amend item?'. You can amend the chart in as many ways as you wish and obtain printouts of all versions of the chart.

A CASHFLOW example

The best way to learn how to use CASHFLOW is to follow this example. To begin with, the Smith family selected April as the 'From' month, and chose a 14 month period. They asked for only eight items, to keep the example fairly simple. The screen cleared and the display showed the column headings and row letters (not the item names) as in the figure. Apart from this, the rows and columns of the table were blank.

They were then asked to key in items A to H. These are the details they keyed in:

Item A: Name: INCOME: Initial value = 0 (more about initial value later). Their income is £550 monthly, until the new year when it will rise to £600. In this example, we are assuming that this is net income after deduction of income tax, national insurance contributions and other amounts deducted at source. The details were entered as follows: In response to 'Period 1–', they keyed '9' (the column number for December). After the 'equals' sign, they keyed the amount '550'. While entering this information, the line of the display went through these stages:

```
Period 1–
Period 1–9          (then press ENTER)
Period 1–9 =
Period 1–9 = 550 (then press ENTER)
```

All that they had to type in was the '9' and the '550'. Typing in a single numerical value after the 'equals' makes the computer put that amount in every column of the period specified when it displays the cashflow later.

After they had keyed '550' and pressed ENTER, and since they had not specified a period up to the end of the cashflow, the computer asked 'Period 10–'. This shows the stages in their next response:

```
Period 10
Period 10–14          (then press ENTER)
Period 10–14 =
Period 10–14 = 600 (then press ENTER)
```

All they had to type was the '14' and the '600'. This completed the entries for Item A. When asked 'ALL OK?', the Smiths replied by keying 'Y'. Note that no figures appear in the cashflow yet.

Item B: Name: HOUSE: Initial value = 0. The expenditure was reckoned as £100 monthly for mortgage repayments, plus £50 in April and October for rates. When they had finished entering the details, the screen showed:

```
    Item B   Name: HOUSE

Initial value (0-9999)  0

    Period 1–1    = 150
    Period 2–6    = 100
    Period 7–7    = 150
    Period 8–12 = 100
    Period 13–13 = 150
    Period 14–14 = 100
```

(If you have more than six periods, data collection is continued at the right side of the screen.)

Item C: Name: CAR: Initial value = 0. The car expenses are £65 monthly until next January. After this the hire purchase ends and it costs only £20 a month. In July there is an extra £30 for a servicing before going on holiday. These details were keyed in just as for Item B above, the periods being 1–3, 4–4, 5–10 and 11–14.

Item D: Name: FOOD: Initial value = 0. This is even simpler, since Mrs Smith estimated to spend £180 monthly.

Item E: Name: FUEL: Initial value = 0. They spend £15 monthly in the summer, and £30 monthly in the winter.

Item F: Name: HOLIDAYS. Having entered the basic items of their cashflow, they now come to the less essential, though very important items. Holidays are financed by setting aside a certain amount each month. They have already accumulated £90 since returning from last year's holiday, so HOLIDAY begins with an initial value of 90. This is the purpose of 'Initial value'. It lets you begin a cashflow with an amount of money in hand. If you want to do a cashflow over a period longer than 14 months, you could start a new cashflow using all the values from the last month of the previous cashflow as 'Initial values' of the new one.

Having keyed in the initial value, they now have to work out how the holiday fund will be calculated each month. The basis for calculation they decided on is to find out how much money they have left from their income after deducting for HOUSE, CAR, FOOD and FUEL. One eighth of this is to be put into the HOLIDAY fund. This amount might be put into a separate account at the bank, or (less safely) kept in a piggy bank. If they represent INCOME by the letter A, HOUSE by B, CAR by C, FOOD by D and FUEL by E, the amount of money left over from income after these expenses is:

$$A-B-C-D-E$$

One eighth of this is:

$$(A-B-C-D-E)/8$$

The amount calculated so far is the amount they put into the holiday fund each month. The amount actually in the fund is that amount plus any amount already there. Since F is the letter referring to the HOLIDAY item, they represent the amount already in the fund by the corresponding lower-case letter 'f'. The complete calculation of the amount in the fund becomes:

$$f+(A-B-C-D-E)/8$$

In words, the amount in the fund is the amount already there, plus an eighth of what is left over from income after deducting essential expenses. This is their holiday saving plan and they must tell the computer about it. Telling the computer is simple, but we must first eliminate the brackets:

$$f + A/8 - B/8 - C/8 - D/8 - E/8$$

They want the computer to work in this way for all months up to July. The display will show:

Period 1–4 = f + A/8 − B/8 − C/8 − D/8 − E/8

It is important not to confuse capital letters and small letters, and to type any spaces. Since the display is in 80-column mode it is easier to make mistakes with certain letters. Look carefully at what you have typed before pressing ENTER.

In August, they go on holiday and happily spend everything saved:

Period 5–5 = 0

Their holiday fund is exhausted (and so are the Smith family!) but, on their return, they start saving for next year's holiday according to the same plan:

$$\text{Period 6–14} = f + A/8 - B/8 - C/8 - D/8 - E/8$$

This is exactly the same as for the pre-holiday period. Note that letter 'f' stands for whatever is in the fund. In month 1, the computer takes the 'initial value' as f. In month 2 and subsequent months it takes the value it has calculated for F in the previous month. Thus any 'small' letter stands for 'initial value' in month 1, and for 'previous month' after that.

Item G: Name: SAVINGS: This is the amount they have in the bank. At the start of the cashflow period, the initial value is £245. The amount in the bank at the end of each month may be calculated as follows:

Last month's balance + income − all expenses

Translated into letters, this becomes:

$$g+A-B-C-D-E-(F-f)-h$$

The first six letters are easily understood. The pair (F − f) are required because the value F always represents the *total* in the holiday fund. The amount actually *put into* the fund in any given month is the difference between the amount in the fund in that month (F) and the amount in the fund in the previous month (f), that is to say (F − f). Letter 'h' refers to expenses on clothing in the previous month, which we shall deal with below. Removing the brackets gives:

$$g+A-B-C-D-E-F+f-h$$

This expression covers all months of the cashflow except August (1–4 and 6–14). In this holiday month, the value of F suddenly becomes zero. Using (F − f) gives a wrong result. Instead they must work out directly what to put into the fund in August (and spend straight away). For August (5–5) the SAVINGS item is calculated as:

$$g+A-B-C-D-E-(A-B-C-D-E)/8-h$$

(F − f) has been replaced by one eighth of income minus essential expenses. Without brackets, this becomes:

$$g+A-B-C-D-E-A/8+B/8+C/8+D/8+E/8-h$$

This is the expression to be typed for August.

Item H: Name: CLOTHES: This might be taken to include other expenses, such as household equipment, and hobbies. The Smiths decide to look at their bank account at the end of the month and spend one fifth of it on clothes (etc.). The formula for this is:

G/5

In making this entry they keyed 0 as the initial value. (Now we can see why 'h' appeared in the calculation for SAVINGS.) Each month they spend one fifth of the amount left in the bank at the end of the previous month (h).

Having keyed in these details they are rewarded by the display shown in Fig. 1.1. Looking first at that crucial item, SAVINGS, they were pleased to see that there would be a substantial increase in the months ahead. There will be a slight fall in October, but nothing to worry about. The amount they can spend on clothing is increasing likewise – perhaps they would not need to spend all of this allocation. The HOLIDAY fund rises to £176 in July. This, plus part of the FOOD and FUEL allowance for August is available for holidaying, plus an eighth of A − B − C − D − E (£24). They will have over £200 to spend. Some of this can immediately be used as a deposit and they book a holiday costing up to that amount, knowing that they can afford it when the time comes. In fact, with such a good rate of saving, perhaps they could spend more. The Smiths can also see that by next April, their holiday fund will hold £245. This is better than the £90 they have this April, indicating the possibility of affording a more expensive holiday next year.

Then a thought occurs – why not sell the old car in July, when it is due to be serviced and buy a new one? Their estimate for having a new car would put up the CAR expenditure to £120 monthly.

In response to 'Amend entry?' they key 'C'. Then enter the following (after keying '0' for 'initial value'):

1–3 = 65
4–14 = 120

Immediately a new cashflow chart is generated, taking this new calculation into account. Since F affects item G directly, and item H indirectly (the amount for clothing depends on how much has been saved), the figures in rows F, G and H are all changed.

Before they have time to study these, another thought occurs to them. Mrs Smith has allowed £180 monthly for food, but there is inflation to think of. If they want to eat as well in 14 months' time as now, she may need to spend more on food each month. Inflation is

```
           Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec   Jan   Feb   Mar   Apr   May
           1     2     3     4     5     6     7     8     9     10    11    12    13    14
A INCOME    550   550   550   550   550   550   550   550   550   600   600   600   600   600
B HOUSE     150   100   100   100   100   100   150   100   100   100   100   100   150   100
C CAR        65    65    65   120   120   120   120   120   120   120   120   120   120   120
D FOOD      181   182   183   184   185   186   187   188   189   190   191   192   193   194
E FUEL       15    15    15    15    15    15    30    30    30    30    30    30    15    15
F HOLIDAY   107   131   154   170     0    16    24    38    52    72    92   112   127   148
G SAVINGS   367   458   530   539   545   549   494   493   491   533   565   590   579   613
H CLOTHES    73    92   106   108   109   110    99    99    98   107   113   118   116   123
```

Fig. 1.2. The cashflow of Fig. 1.1 amended to allow for buying a new car and for inflation of food prices.

about 6% per annum. Roughly, this is a half per cent increase monthly (actually it is less, but the difference is not important). So she begins the food calculation with an initial value of £180 and inflates this by a half per cent each month. Initial value is 180 and the calculation of D is:

d*1.005

Each month the amount of the previous month is multiplied by 1.005. Once again the figures in the table change. This time, rows D, F, G and H are affected. Figure 1.2 shows the results of the amendments to CAR and FOOD. FOOD increases by £1 each month, which is less worrying than she thought it might be. Happily, it looks as if the Smith family can afford the new car. SAVINGS are not as great as was predicted by Fig. 1.1, but there is still an overall increase. The drop in savings in October, November and December might have been very worrying if they had not done a cashflow and were relying only on monthly bank statements received at the ends of those months. CASHFLOW shows that the situation will improve when Mr Smith gets the rise in pay next January. Rather than wait for the rise before buying the car, they bought the car in time for this year's holiday, knowing that they can afford it in the long run. It is for this kind of long-range planning that a cashflow forecast is most helpful. But remember that the forecast is only as reliable as the information you feed into the computer. If you have forgotten to tell it that you are spending £100 a month on entertaining your friends to parties, the financial future may be less rosy that it looks on the chart. Also an unexpected expense, such as wind damage to your roof, could (unless you are insured) reduce your SAVINGS balance considerably. Keep funds in reserve for emergencies, and insure yourself against those risks you cannot afford to meet.

Good things have to be paid for, a fact that is shown on the chart. The purchase of a new car means that the Smiths must spend less on their holiday, and on clothes. They might prefer to put a bigger portion of their surplus cash into the holiday fund, save less and spend less on clothing. They have only to amend item F to see what effect this will have in the next 14 months. Perhaps 'f + A/6 − B/6 − C/6 − D/6 − E/6' would be a better basis for setting aside their holiday money. Perhaps 'G/8' would provide enough for clothing. Having done this, they might be able to spend even more on the car! You can continue indefinitely in this way, trying one thing, then another, until you have planned your cashflow to your satisfaction.

Other calculations

The example has used various kinds of 'calculations'. These have ranged from one or more constants (items A to E), a simple multiplication or division by a given factor (item H, and the inflation-indexed calculation for D) to a long succession of additions and subtractions (items F and G). The example covers most of the kinds of calculation that you are likely to want to use. But beware! Syntax errors or invalid statements made when typing in the calculation line could cause the program to crash. Another point worth noting is that only integers are used in the results. Negative values are allowed; this is when you start to think about asking your bank manager for an overdraft! If possible, try not to let your cashflow produce values bigger than 9999, as this will upset the neat rows and columns of the display. If your cashflow runs to five-figure amounts, you can afford to buy more expensive spreadsheet software.

We have assumed that you will use only reasonably sensible calculations in this program. The routine works only with the operators '+', '–', '*' and '/' (divide). It gives priority to multiplication and division (working from left to right) and then performs additions and subtractions. It does *not* accept brackets. The range of possibilities is great. We prefer not to restrict what you key in, even though there is the risk of your keying in an unacceptable item. In this way, this fairly short program has the greatest possible flexibility in use.

One final point to consider is that the program works column by column, passing down the columns as it goes. Consequently, values in a column are updated from the top downward. It may not be sensible for a calculation of a given item to include a capital letter relating to an item further down the list. This later item could possibly be recalculated as the program proceeds, making nonsense of the result worked out earlier. If you know from what you have entered that a value further down the column will *not* change then including it in an item further up the list is allowable. It is always in order to use lower-case letters in calculations, no matter what their position in the list may be.

Keying in

A number of the 'strings' have spaces in them. There are examples in lines 50, 70, 420 and elsewhere. Take care to key in the correct number of spaces, otherwise an untidy display may result.

The variable in lines 100, 140, 180, 410, 530 and 670 is ‘NI’ letter ‘en’, letter ‘eye’).

Program design

20–30 Initialising mode, defining integers, arrays, and a variable.

40–100 Queries relating to the scope of the cashflow.

110–130 Displaying the table with row and column labels.

140–160 Entering information until all correct.

170–380 Evaluating values in array D, using expressions stored in D$.

390–480 Clear bottom of screen and input data to amend calculations.

490–590 Printout routine.

610–650 Subroutine to display a query message, input numbers between preset upper and lower limits, the message being tabbed to a given screen line.

660–680 Subroutine to display letters on left side of screen.

690–930 Subroutine to input all information relating to a given item.

940–960 DATA statement containing sequence of month names for column headings.

The program

```
10 REM ** CASHFLOW **
20 MODE 2:DEFINT C,D,J,P,Z
30 DIM D(14,12),D$(14,12),M$(14),P(20),V
(20):FA=0
40 PRINT TAB(35)"CASHFLOW"
50 M$="Starting from which month? (1-12)
 ":LL=1:UL=12:TB=5:GOSUB 610
60 FOR J=1 TO M-1:READ X$:NEXT
70 M$="For how many months? (2-14)   ":LL
=2:UL=14:TB=7:GOSUB 610
80 NM=M:FOR J=1 TO NM:READ X$:M$(J)=X$:N
EXT
90 M$="How many items?  (1-12)":LL=1:UL=
12:TB=9:GOSUB 610
100 NI=M:CLS
```

```
110 FOR J=1 TO NM:LOCATE (6+5*J),1:PRINT
 M$(J);:NEXT
120 FOR J=1 TO NM:LOCATE (6+5*J),2:PRINT
 J;:NEXT
130 GOSUB 660
140 FOR J=1 TO NI
150 GOSUB 690
160 NEXT
170 FOR Z=1 TO NM
180 FOR ZZ=1 TO NI
190 FOR JZ=1 TO 20:P(JZ)=0:V(JZ)=0:NEXT
200 D$=D$(Z,ZZ):JN=0:JP=1:JV=1
210 FOR JD=1 TO LEN(D$):C=ASC(MID$(D$,JD
,1))
220 IF JN>0 THEN JN=JN-1:IF JN>0 THEN 27
0
230 IF C=42 OR C=43 OR C=45 OR C=47 THEN
 P(JP)=C:JP=JP+1
240 IF C>64 AND C<77 THEN V(JV)=D(Z,C-64
):JV=JV+1
250 IF C>96 AND C<109 THEN V(JV)=D(Z-1,C
-96):JV=JV+1
260 IF C>47 AND C<58 THEN V(JV)=VAL(RIGH
T$(D$,LEN(D$)-JD+1)):JN=LEN(STR$(V(JV)))
-2:JV=JV+1
270 NEXT
280 FOR P=1 TO JP
290 IF P(P)=42 THEN V(P)=V(P)*V(P+1):V(P
+1)=0:P(P)=43
300 IF P(P)=47 THEN V(P)=V(P)/V(P+1):V(P
+1)=0:P(P)=43
310 NEXT
320 FOR P=1 TO JP
330 IF P(P)=43 THEN V(1)=V(1)+V(P+1)
340 IF P(P)=45 THEN V(1)=V(1)-V(P+1)
350 NEXT
360 D(Z,ZZ)=CINT(V(1))
370 LOCATE (5+5*Z),(2+ZZ):PRINT USING "#
###";D(Z,ZZ)
380 NEXT:NEXT
390 LOCATE 1,16:PRINT CHR$(20)
400 FA=1
410 J=0:WHILE (J<1 OR J>NI) AND J<>16
420 LOCATE 1,16:INPUT "Amend item?  (Key
 letter)  ",L$
```

```
430 IF A$="" THEN 420
440 J=ASC(L$)-64:IF J>32 THEN J=J-32
450 WEND
460 IF J=16 THEN 490
470 GOSUB 690
480 GOTO 170
490 PRINT #8, SPACE$(10);
500 FOR J=1 TO NM:PRINT #8,"  ";M$(J);:N
EXT:PRINT #8, CHR$(13)
510 PRINT #8, SPACE$(10);
520 FOR J=1 TO NM:PRINT #8, USING "#####
";J;:NEXT:PRINT #8, CHR$(13)
530 PRINT #8, CHR$(13):FOR J=1 TO NI

540 D$=LEFT$(D$(0,J)+STRING$(10," "),10)
550 PRINT #8, CHR$(64+J)" "D$;
560 FOR K=1 TO NM:PRINT #8, USING "#####
";D(K,J);:NEXT
570 PRINT #8,CHR$(13)
580 NEXT
590 GOTO 170
600 END
610 M=-1:WHILE M<LL OR M>UL
620 LOCATE 1,TB:PRINT M$;:INPUT" ",A$
630 M=VAL(A$)
640 WEND
650 RETURN
660 LOCATE 1,3
670 FOR J=1 TO NI:PRINT CHR$(J+64):NEXT
680 RETURN
690 A$="":WHILE A$<>"Y"
700 IF FA=0 THEN LOCATE 2,(J+2):PRINT"
     "
710 LOCATE 1,16:PRINT CHR$(20)
720 LOCATE 4,16:PRINT"Item ";CHR$(J+64);

730 IF FA=1 THEN LOCATE 15,16:PRINT D$(0
,J):GOTO 770
740 LOCATE 15,16:INPUT "Name: ",D$
750 D$(0,J)=LEFT$(D$,7)
760 LOCATE 3,(J+2):PRINT D$(0,J)
770 M$="Initial value (0-9999)  ":LL=0:U
L=9999:TB=17:GOSUB 610
780 D%(0,J)=M
790 X=2:Y=17:FM=1:LM=0
```

```
800 WHILE LM<>NM
810 Y=Y+1:IF Y=24 THEN X=41:Y=16
820 WHILE LM<FM OR LM>NM
830 LOCATE X,Y:PRINT "Period "FM;:INPUT"
- ",LM$
840 LM=VAL(LM$)
850 WEND
860 LOCATE (X+15),Y:INPUT "= ",E$
870 FOR K=FM TO LM:D$(K,J)=E$:NEXT
880 FM=LM+1
890 WEND
900 LOCATE 66,Y:PRINT "All OK? (Y/N)";

910 A$=UPPER$(INKEY$)
920 IF A$="" THEN 910
930 WEND:RETURN
940 DATA Jan,Feb,Mar,Apr,May,Jun,Jul,Aug
,Sep,Oct,Nov,Dec
950 DATA Jan,Feb,Mar,Apr,May,Jun,Jul,Aug
,Sep,Oct,Nov,Dec
960 DATA Jan,Feb
```

Variations

The program could easily be adapted to handle data of other kinds, including non-integer values. You would need to change the column headings, several of the messages and probably the number of rows and columns.

2

Gottit!

Whenever you are working for exams, be they GCE, CSE, BEC, TEC, the driving test or simply first-form geography, it is always useful to have a friend to help test your knowledge. The micro can remember dozens of questions and their answers. It fires the questions at you in random order, so that you never know exactly what it is going to ask you next. It also keeps a score of your successes and helps you when you go wrong.

Using the program

The program consists of two sections. One is for use in setting the questions; you type in the questions for the micro to ask you later. You also type in the correct answers. At this stage you are using your books and notes to make certain that the answers are correct. The other section of the program is that in which the computer chooses questions in random order from those you have typed in and then asks you to key in the answers. At this stage you have put your books and notes away and hope to be able to recall the correct answers from memory to match those in the computer.

When the program is first run, you are asked 'Set questions or answer them?'. Key in 'S' or 'A', according to which section of the program you want to use. If you type 'S' to set the questions, the screen becomes black and you are then asked 'Topic?'. Respond to this by typing a name which indicates what the questions will be about. Examples of names are 'MAPS', 'PHYSICS', 'BOOK KEEPING', 'DRIVING TEST' and 'LIFE SAVING'. Topic names must not be more than sixteen characters long.

The computer then asks you to type in each question, and then its answer. The program holds up to 125 questions. Each question and answer may be up to 255 characters long (just over six lines on the screen). However, if you were to make all the questions and answers as long as this, the micro would run out of memory. The average length of questions and answers should be about forty-five characters (a little more than one screen line). Actually you may need to make the questions a little longer than this, and the answers rather shorter. It is best to keep the answers as short as possible, to make it easier for the computer to match the stored answer against the one you type in later. Single-word answers are the best of all. Answers can also be equations (e.g. $V = IR$) or chemical formulae (e.g. H_2SO_4). One very important point – *do not use commas* when typing questions or answers.

If answers are to be as short as possible, the questions must be designed to be correctly answered by one-word or few-word answers. The task of designing questions to cover your topic fully, and working out the best short answers for typing into the computer is, in itself, a tremendous aid to learning. It demands mental activity and is far more effective than simply reading through a textbook or a set of notes. You will probably find that, by the time you have worked out the questions and answers, you have already learnt almost all you need to know about the topic. How useful your computer-stored questions and answers will be for revision!

After you have typed in each question and its answer, you are asked 'More?'. If you have more questions to type in, key 'Y'. Provided that you have not already typed in 125 questions, you will then be asked to key in the next question and its answer.

When you have finished typing questions and answers, key 'N'. The message appears on the screen: 'Press REC and PLAY then any key:'. This is for you to save the questions and answers on tape, so that the computer can test you in a few days' time. Place a tape in the recorder and fast-forward or rewind it to a suitable place. Press REC and PLAY on the recorder. Then press any key on the micro. If you have typed many long questions, saving may take up to five minutes.

When saving is complete, you are asked if you require a backup copy. If so, key 'Y' and repeat the procedure above. If not, key 'N' and the program ends.

When you RUN the program to test your knowledge, key 'A' in answer to the question 'Set questions or answer them?'. The screen changes to white, with red text. As in the other section of the program, you will be asked 'Topic?'. Put the tape for this topic in the recorder. It might be one you have just recorded, or perhaps one recorded several days or weeks ago. It is a good idea to make these tapes while you are working on a course and keep them ready for revision just before the exams.

When you have keyed in the name of the topic, press ENTER. Press PLAY on the recorder, then any key. After a period of up to five minutes, depending on how many questions there are, the screen clears and the first question is displayed. Above it, you will see your score, which is up-dated each time a question is asked. To begin with it says 'your score is 0 out of 0'.

Read the question, then type in the answer. If your answer is *exactly* correct, the message 'CORRECT' appears and your score is incremented by 2. The micro will only count an answer as correct if it is typed exactly the same as the answer you typed in when setting the questions. However, if you type in an answer that is partly correct, the computer may still be able to give you a single mark for a good attempt. It displays the message 'ALMOST RIGHT – THE EXACT ANSWER IS:', followed by the exact answer. If you are not sure of the whole answer, but think you know part of it, type in the part which you are sure about. A single word may score a mark for you. It is possible to cheat by typing in one or two letters, but remember you are cheating yourself, not the computer!

If the micro cannot match your answer against any part of the correct answer, and cannot match any part of the correct answer against your answer, it displays the message 'WRONG – THE CORRECT ANSWER IS:' followed by the correct answer.

After each question has been set and the answer assessed by the computer, you are asked 'More questions?'. If you key 'Y', the computer asks you another question. If you key 'N', you are asked if you want another topic. This gives you the chance to load another tape. If you do not choose this option, the program ends.

Keying in

The only point about keying in this program is to make sure you type the correct number of 'PRINT's in the lines where they occur.

Program design

20–70 Initialising and setting variables. Lines 30–50 set aside memory for saving and loading.

80–100	Setting or answering?
110–190	Setting questions and answers.
200–310	Saving questions and answers.
320–410	Loading questions and answers.
420–530	Asking random questions and receiving user's answers.
540–580	Option to load another tape.
590–600	Subroutine to get Y/N responses.

The program

```
10 REM ** GOTTIT! **
20 MODE 1
30 OPENOUT "AMSTRAD"
40 MEMORY HIMEM-1
50 CLOSEOUT
60 DIM Q$(125,1):N=1
70 INK 0,1:INK 1,24
80 CLS:PRINT "Set questions or answer th
em? (S/A)"
90 A$=UPPER$(INKEY$):IF A$<>"S" AND A$<>
"A" THEN 90
100 IF A$="A" THEN 320
110 INK 0,0:PRINT:PRINT
120 INPUT "Topic ";T$
130 CLS:PRINT "Question No. ";N":"
140 PRINT:INPUT Q$(N,0)
150 PRINT:PRINT:PRINT "Answer:"
160 PRINT:INPUT Q$(N,1)
170 PRINT:PRINT:PRINT:PRINT "More?"
180 GOSUB 590
190 IF A$="Y" THEN N=N+1:IF N<126 THEN 1
30
200 OPENOUT T$
210 PRINT #9,N
220 FOR J=1 TO N
230 FOR K=0 TO 1
240 PRINT #9,Q$(J,K)
250 NEXT:NEXT
260 CLOSEOUT
270 PRINT "Back-up? (Y/N)"
280 GOSUB 590
290 IF A$="Y" THEN 200
300 PRINT "Recording finished"
310 END
320 NC=0:NQ=0
```

```
330 INK 0,13:INK 1,3
340 CLS:INPUT "Topic";T$
350 OPENIN T$
360 INPUT #9,N
370 FOR J=1 TO N
380 FOR K=0 TO 1
390 INPUT #9,;Q$(J,K)
400 NEXT:NEXT
410 CLOSEIN
420 CLS:PRINT "Your score is ";NC;" out
of ";NQ
430 NQ=NQ+1:PRINT:PRINT "Question";NQ
440 R=INT(RND(1)*N)+1:PRINT:PRINT:PRINT
;Q$(R,0)
450 PRINT:PRINT:PRINT "Answer"
460 PRINT:PRINT:INPUT answer$
470 IF answer$=Q$(R,1) THEN PRINT:PRINT:
PRINT "CORRECT":NC=NC+2:GOTO 510
480 IF INSTR(answer$,Q$(R,1)) OR INSTR(Q
$(R,1),answer$) AND answer$<>"" THEN PRI
NT:PRINT "ALMOST RIGHT - THE EXACT ANSWE
R IS:":NC=NC
+1:GOTO 500
490 PRINT:PRINT:PRINT "WRONG - THE CORRE
CT ANSWER IS:"
500 PRINT:PRINT:PRINT Q$(R,1)
510 PRINT:PRINT:PRINT "More questions? (
Y/N)
520 GOSUB 590
530 IF A$="Y" THEN 420
540 CLS:PRINT:PRINT:PRINT "Another topic
? (Y/N)"
550 GOSUB 590
560 IF A$="Y" THEN 320
570 PRINT:PRINT:PRINT:PRINT "GOTTIT! FIN
ISHED"
580 END
590 A$=UPPER$(INKEY$):IF A$<>"Y" AND A$<
>"N" THEN 590
600 RETURN
```

Variations

You may prefer a strict test of your knowledge with no 'almost correct' answers allowed. If so, delete line 480. Another variation is for the computer to let you have another attempt at answering the question when you succeed in typing a partly correct answer. You might then gradually work out what the correct answer is by making successive attempts at it. If you would prefer the program to operate in this way, substitute this line for line 480:

```
480 IF INSTR(answer$, Q$(R, 1)) OR INSTR(Q$(R, 1),
answer$) THEN PRINT "ALMOST RIGHT – TRY
AGAIN!":GOTO 460
```

3

Stock Take

Here is a program with thousands of applications. In the home, the workshop or the office, use it for keeping an inventory of ingredients, equipment, materials or parts. Not only does it display a complete list, it will prepare a list of all items for which present stocks are below the minimum level. Deciding what you need to buy to maintain your stock is just a matter of noting down the details from the screen as the micro presents them to you. If your stock is of a perishable kind, such as the smoked salmon in the deep-freeze, or the packets of photographic paper on your dark-room shelf, the program can identify old stock by displaying a list of all items entered before a selected date.

Several people can use the program, each having their own tape or tapes on which their stock lists are held. There can be a tape for the cook, listing the contents of the food cupboard and the deep-freeze, and a components tape for the electronic wizard of the household. The gardener of the family can have a tape for stocks of fertilizer, seeds, flowerpots and other consumable items. The home brewer can list stocks of sugar, malts, hops, dried fruits and flowers, as well as of the finished products. The DIY expert can list stocks of timber, nails, screws and all the countless other items needed in the efficient home workshop. It is very useful to keep an inventory of the furniture and furnishings for insurance purposes. The program can hold details of dates of purchase and the cost of all such items, making it very easy to be certain that your insurance cover is adequate year by year.

Although the program has many features built in, and is ready to use, it is designed so that it can be tailored to your requirements, as explained in the *Variations* section of this chapter.

There is room for up to 500 records on any one stock list.

STOCK TAKE records

Each record on the stock list consists of eight items of information:

(1) *Code:* a word up to four letters long, or a group of up to four letters and numerals. The purpose of this is to identify certain categories of stock quickly. For example, a stock list of electronic components could have the code RES for all items that referred to stocks of resistors. In a kitchen stock list, you might use the code DF for all deep-freeze stock, to distinguish it from stocks of other kinds. Keying in DF would then make the computer display a list of all deep-frozen stocks, as explained later.

(2) *Description:* words or codes to identify each item exactly. Up to ten characters are allowed here. This number has been restricted in order to fit a complete record into one line of a 40-column screen. However, it is usually easy to shorten words or use spare parts numbers so as to specify the nature of the item exactly within the space allowed.

(3) *Date:* this can be the date of purchase, the date on which the stock quantity was last amended, or any other date which suits your system of stock-taking. You can enter the date in any format you choose, provided that it has no more than six characters. Thus the 3rd September 1985 could be keyed in as '030985', '30985', '090385', '03SE85', '3 SEPT', or '3/9'. The only point to remember is that the 'Old List' routine works only with dates entered according to the first format above (DDMMYY).

(4) *Quantity:* this is the number of items currently held in stock. The program accepts any number (including decimal fractions) up to 9999. It is best not to use numbers with more than two places of decimals, for this upsets the arrangement of the display. The same points apply to numbers entered in items (5) and (6) below.

(5) *Minimum quantity:* this is the minimum quantity that you wish to keep in stock. When the stock quantity has fallen to this minimum or below it, you need to buy more of the item. The 'Order List' option (see later) tells you what items have to be replenished.

(6) *Re-order quantity:* this is the quantity of each item which should be bought when the stock quantity falls below the minimum stock quantity.

(7) and (8) *Fields 1 and 2:* these give you two spare columns in the

stock list in which you can record any information you wish. One of the fields might give the latest price of the item; another might say from whom the item was purchased.

Using the program

Before you use the program for the first time, or when you want to begin a new stock list, you need to prepare a tape on which the stock list is to be recorded. Use the START LIST program for this. All this does is to put an 'empty' list on to tape, ready for use by the STOCK TAKE program.

Run the START LIST program. When it asks you 'Name of list?', type in the name you have decided to give to your stock list (up to 16 characters). It might be 'CANNED FOOD', 'GARDEN' or 'SPARES'. Write the chosen name on the cassette label, then put the cassette into the tape-recorder. The screen will show the message 'Press REC and PLAY then any key'. Wind the tape to the beginning, if necessary, so that it is ready for making a recording. Press the RECORD and PLAY buttons on the recorder, then press any key on the micro. Creating the 'empty' list will take only a few seconds. Then the message 'Empty list ready' appears on the screen, and the program has finished.

Now load the STOCK TAKE program. When you RUN this program, you are first asked to key in the name of the list. This is the name you gave it when you used the START LIST program. Key in its name, and press ENTER. Remove the STOCK TAKE tape and replace it with the 'list' tape. Rewind the tape to the beginning, if necessary, then press the PLAY button on the recorder, then any key. If you are starting a new list, loading it takes only a few seconds. If, on the other hand, you are loading a list which already holds many items, it will take several minutes.

When loading is complete, the screen clears and the menu is displayed. This allows you a choice of any of six options:

(1) *New entry:* Use this to take a new item on to the list. You are asked to key in the eight pieces of information required. Refer to the headings (under *STOCK TAKE records*) to remind yourself of these, and what restrictions (e.g. number of characters allowed) each has. Press ENTER after entering each piece of information. You do not have to make an entry for each heading. If there is nothing to enter, just press ENTER. When entry is complete, the screen

clears and displays everything you have just typed, in the form in which it will be taken on to the list. This gives you a chance to look for typing errors. Look to see if you have typed in more than the allowed number of characters, and so have 'lost' the end of an entry. If all is correct, key 'Y' and you will be returned to the menu. Otherwise, type 'N' to cancel the entry and re-enter the details correctly.

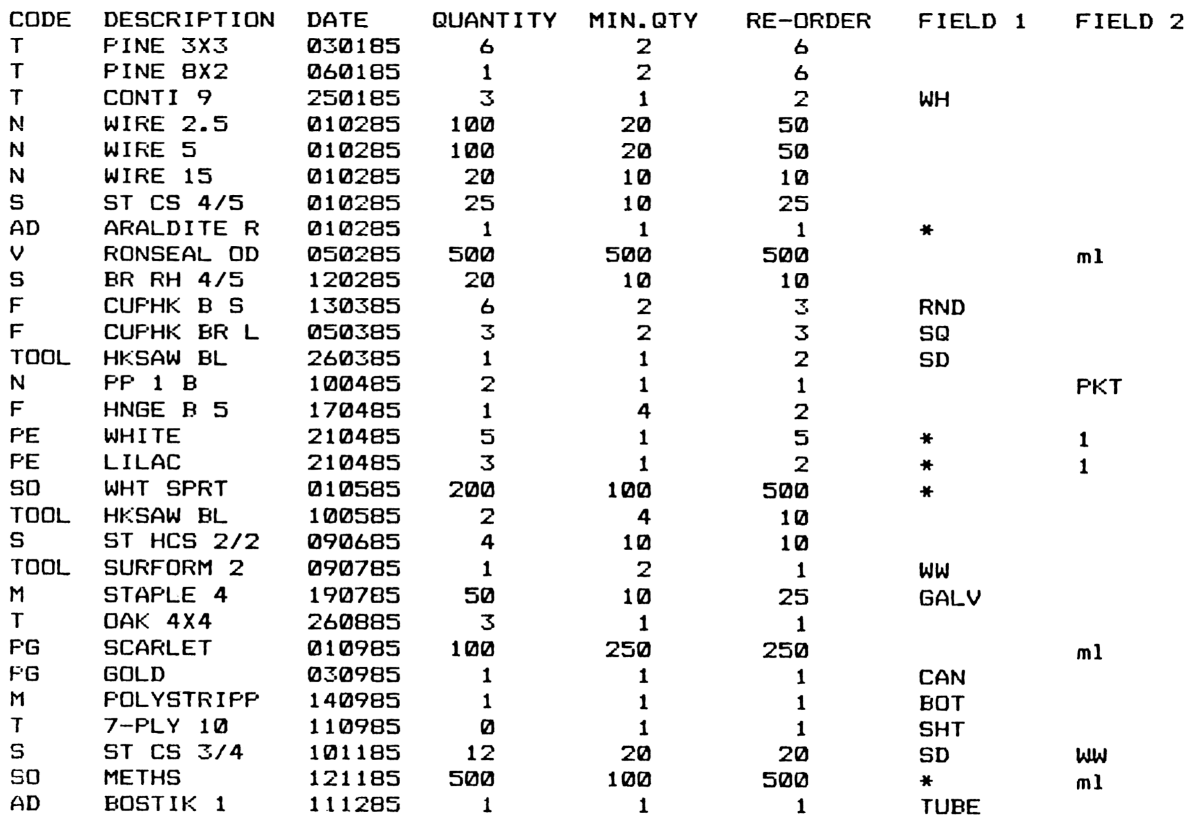
There is no indication in the program of the number of records currently in the list. Up to 500 records are allowed. When the number held exceeds 495 the micro 'beeps' at you each time you choose this option. If there are more than 500 records, it returns you to the menu.

(2) *Display stock list:* This shows you the whole list from beginning to end, in the order in which the records were entered. Figure 3.1 is an example of a list compiled by a DIY enthusiast. The items are coded, T standing for timber, N for nails, S for screws, PE for emulsion paint and so on. Alternate columns of information are displayed on the screen in red-on-yellow, or yellow-on-red, to make them easy to distinguish when packed with information. If you have more than a screenful of records you can make the micro wait for you to read them. Press ESC, then press the space-bar to view the next screenful.

When the whole list has been displayed, the message at the bottom of the screen is 'Space bar = Continue:P=PRINT'. Pressing the space-bar returns you to the menu. Pressing 'P' prints out the list on your printer.

(3) *Display/Amend entry:* This option finds any single entry or group of similar entries without the need to view the whole stock list. An entry may be defined by specifying its code or its description. You are asked to type in both of these, but need type in only one. Press ENTER. If you specify the code, you will be shown only the records which have this code. For example, with the list shown in Fig. 3.1, the code 'T' would cause all entries relating to timber to be displayed. If you type a description only, items having that description will be displayed. Since each description should be unique, you will probably be shown only one item.

The records are displayed one at a time to give you the opportunity of deciding whether to amend them. If you do not want to amend a record, press the space-bar. If there are further records conforming to the code or description, the next one will then be displayed. If there are no more, the message 'END OF SEARCH'



```
CODE  DESCRIPTION   DATE     QUANTITY  MIN.QTY   RE-ORDER  FIELD 1   FIELD 2
T     PINE 3X3      030185      6         2         6
T     PINE 8X2      060185      1         2         6
T     CONTI 9       250185      3         1         2      WH
N     WIRE 2.5      010285    100        20        50
N     WIRE 5        010285    100        20        50
N     WIRE 15       010285     20        10        10
S     ST CS 4/5     010285     25        10        25
AD    ARALDITE R    010285      1         1         1      *
V     RONSEAL OD    050285    500       500       500                ml
S     BR RH 4/5     120285     20        10        10
F     CUPHK B S     130385      6         2         3      RND
F     CUPHK BR L    050385      3         2         3      SQ
TOOL  HKSAW BL      260385      1         1         2      SD
N     PP 1 B        100485      2         1         1                PKT
F     HNGE B 5      170485      1         4         2
PE    WHITE         210485      5         1         5      *         1
PE    LILAC         210485      3         1         2      *         1
SO    WHT SPRT      010585    200       100       500      *
TOOL  HKSAW BL      100585      2         4        10
S     ST HCS 2/2    090685      4        10        10
TOOL  SURFORM 2     090785      1         2         1      WW
M     STAPLE 4      190785     50        10        25      GALV
T     OAK 4X4       260885      3         1         1
PG    SCARLET       010985    100       250       250                ml
PG    GOLD          030985      1         1         1      CAN
M     POLYSTRIPP    140985      1         1         1      BOT
T     7-PLY 10      110985      0         1         1      SHT
S     ST CS 3/4     101185     12        20        20      SD        WW
SO    METHS         121185    500       100       500      *         ml
AD    BOSTIK 1      111285      1         1         1      TUBE
```

Fig. 3.1. Complete stock list.

is displayed. Press the space-bar to return to the menu.

If you decide to amend a record you can either amend the figure in the 'Quantity' column only (key 'Q'), or one or more of the other details (key 'A'). Typing 'Q' provides a quick way of altering the quantities in records as the stock decreases, or when new purchases are made. When you amend only the stock quantity, the screen clears, and you are reminded what the present quantity is. You are then asked to key in the new quantity. After you have pressed ENTER, you will be shown the next record conforming to your specification, or 'END OF SEARCH'.

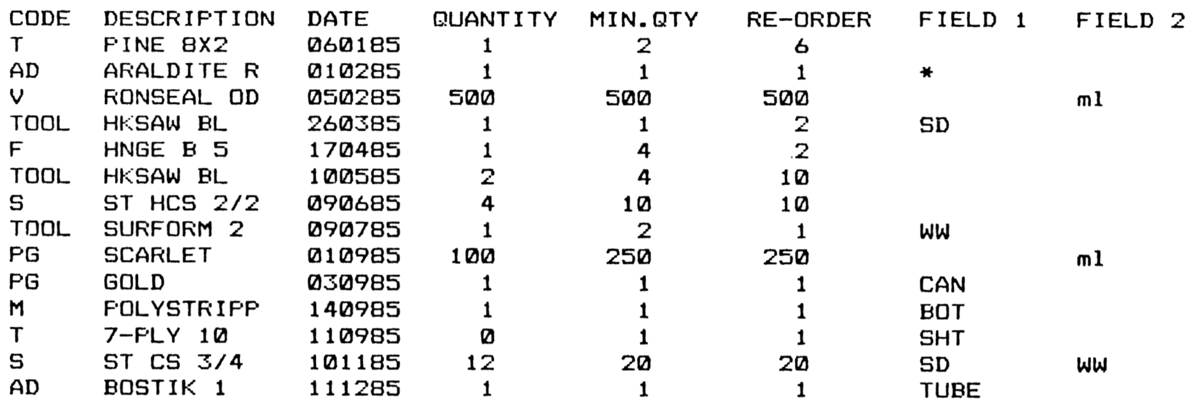
If you decide to amend other parts of the record, you are asked to follow the same procedure as under option 1. All details have to be typed in again, with corrections and amendments. The original version of the record remains on the screen while you type, to allow you to copy in details that are not to be changed. This option can be used to replace an unwanted record with a completely new one.

(4) *Order list:* Selecting this option makes the computer scan the stock list, displaying all records for which the quantity of stock is equal to or less than the minimum stock quantity. Figure 3.2 shows the Order List produced from the stock list of Fig. 3.1. If there are many such records, press ESC to halt the display. Press the space-bar to view the next screenful. When the whole list has been displayed, press 'P' to obtain a printout.

(5) *Old list:* Before the computer searches the list, you are asked to key in the date to be used in the search. This could be the date of purchase of stock, or the 'best before' date of stored foodstuffs. The date is specified by typing in the month (1–12) and the year (0–99). The computer then looks through the list and displays all records for the months up to and including the month specified. Figure 3.3 shows the Old List obtained by selecting February 1985 as the date. The *Variations* section explains how to modify the program to work with day-and-month dates instead of month-and-year dates. This would be more suitable for stores such as foodstuffs, which have a relatively short shelf-life.

As with item (4), press ESC to halt the display, if the list is long. Press the space-bar to continue. Press 'P' to obtain a printout.

(6) *Finish:* this option must be used every time you have finished a



```
CODE  DESCRIPTION  DATE    QUANTITY  MIN.QTY   RE-ORDER   FIELD 1   FIELD 2
T     PINE 8X2     060185      1         2          6
AD    ARALDITE R   010285      1         1          1      *
V     RONSEAL OD   050285    500       500        500                ml
TOOL  HKSAW BL     260385      1         1          2      SD
F     HNGE B 5     170485      1         4          2
TOOL  HKSAW BL     100585      2         4         10
S     ST HCS 2/2   090685      4        10         10
TOOL  SURFORM 2    090785      1         2          1      WW
PG    SCARLET      010985    100       250        250                ml
PG    GOLD         030985      1         1          1      CAN
M     POLYSTRIPP   140985      1         1          1      BOT
T     7-PLY 10     110985      0         1          1      SHT
S     ST CS 3/4    101185     12        20         20      SD        WW
AD    BOSTIK 1     111285      1         1          1      TUBE
```

Fig. 3.2. Order list.

```
CODE  DESCRIPTION  DATE    QUANTITY  MIN.QTY   RE-ORDER   FIELD 1   FIELD 2
T     PINE 3X3     030185     6         2         6
T     PINE 8X2     060185     1         2         6
T     CONTI 9      250185     3         1         2       WH
N     WIRE 2.5     010285   100        20        50
N     WIRE 5       010285   100        20        50
N     WIRE 15      010285    20        10        10
S     ST CS 4/5    010285    25        10        25
AD    ARALDITE R   010285     1         1         1       *
V     RONSEAL OD   050285   500       500       500                 ml
S     BR RH 4/5    120285    20        10        10
```

Fig. 3.3. Old list.

session with the program, in which you have used options 1 or 2 to start new records, or to alter existing ones. As soon as you have selected option 6, the screen clears. The message 'Press REC and PLAY then any key' appears. Rewind the data tape to the place at which the recorded list begins. Press the RECORD and PLAY buttons on the tape-recorder. Then press any key on the micro. A long list may take several minutes to save. When this is complete, the message 'BACKUP' appears. To take a backup copy, key 'Y' and repeat the procedure above. If you press 'N' the message 'STOCK TAKE FINISHED' is displayed.

Keying in

Note the spaces on either side of each column heading in line 1270, except for CODE. The sets of double quotes in lines 1290 and 1300 all have four spaces between them, except for the second set on line 1290, which has ten spaces.

Program design

20–70 Initialising Mode, arrays and colours. Lines 30 to 50 set aside memory for saving and loading.

80–180 Loading data from tape.
190–300 Menu and directing to options.
310–330 New entry routine.
340–430 Display stock list.
440–650 Display/amend entry.
660–790 Order list.
800–840 Old list.
850–990 Finish.
1000–1020 Subroutine to accept 'Y' or 'N'.
1030–1260 Subroutine for inputting data and confirming it.
1270–1280 Subroutine to display column headings.
1290–1310 Subroutine to display details.
1320–1330 Subroutine to print column headings.
1340–1350 Subroutine to print details of one record in the stocklist.
1360–1380 Subroutine to wait for space-bar.

The program (START LIST)

```
10 REM ** START LIST **
20 CLS:INPUT "Name of list";T$
30 OPENOUT T$
40 PRINT #9,0:CLOSEOUT
50 PRINT "Empty list ready"
60 END
```

The program (STOCK TAKE)

```
10 REM ** STOCK TAKE **
20 MODE 1
30 OPENOUT "AMSTRAD"
40 MEMORY HIMEM-1
50 CLOSEOUT
60 DIM D$(500,4),D%(500,2)
70 INK 1,6:INK 0,24
80 CLS:INPUT "Name of list";T$
90 FP=0
100 OPENIN T$
110 INPUT #9,N
120 IF N=0 THEN 180
130 FOR J=1 TO N
140 FOR K=0 TO 4
150 INPUT#9,D$(J,K):NEXT
160 FOR K=0 TO 2
170 INPUT#9,D%(J,K):NEXT:NEXT
180 CLOSEIN
190 CLS:LOCATE 13,1:PRINT "**STOCK TAKE*
*"
200 LOCATE 15,2:PRINT "__________"
210 LOCATE 7,4:PRINT "1)   NEW ENTRY"
220 LOCATE 7,6:PRINT "2)   DISPLAY STOCK
 LIST"
230 LOCATE 7,8:PRINT "3)   DISPLAY/AMEND
 ENTRY"
240 LOCATE 7,10:PRINT "4)   ORDER LIST"
250 LOCATE 7,12:PRINT "5)   OLD STOCK LI
ST"
260 LOCATE 7,14:PRINT "6)   FINISH"
270 LOCATE 5,20:PRINT "<SELECT BY PRESSI
NG NUMBERED KEY>"
280 A$=UPPER$(INKEY$):A=VAL(A$)
290 IF A<1 OR A>6 THEN 280
```

```
300 CLS: ON A GOTO 310,340,440,660,800,8
50
310 N=N+1:J=N:IF N>495 THEN PRINT CHR$(7
):IF N>500 THEN N=500:GOTO 190
320 GOSUB 1030
330 GOTO 190
340 CLS:PRINT T$" STOCK LIST"
350 GOSUB 1270
360 FOR J=1 TO N
370 IF FP=1 THEN GOSUB 1340:GOTO 390
380 GOSUB 1290
390 NEXT
400 PRINT:PRINT:PRINT:PRINT " < SPACE BA
R = CONTINUE : P = PRINT >"
410 A$=UPPER$(INKEY$):IF A$<>"P" AND A$<
>" " THEN 410
420 FP=0:IF A$="P" THEN GOSUB 1320:FP=1:
GOTO 360
430 GOTO 190
440 CLS:PRINT "DISPLAY ENTRY"
450 PRINT:INPUT "CODE ";C$
460 PRINT:INPUT "DESCRIPTION ";D$
470 CLS:GOSUB 1270
480 FOR J=1 TO N
490 IF C$<>D$(J,0) AND D$<>D$(J,1) THEN
580
500 GOSUB 1290
510 PRINT:PRINT:PRINT:PRINT " <SPACE=CON
TINUE : Q=QTY : A=AMEND >":PRINT
520 A$=UPPER$(INKEY$):IF A$<>" " AND A$<
>"Q" AND A$<>"A" THEN 520
530 IF A$=" " THEN 580
540 IF A$="Q" THEN 620
550 PRINT:PRINT:PRINT:GOSUB 1030
560 CLS:GOSUB 1270
570 GOSUB 1290
580 NEXT
590 PRINT:PRINT:PRINT:PRINT TAB(10)"**EN
D OF SEARCH**"
600 GOSUB 1360
610 GOTO 190
620 CLS:PRINT:PRINT:PRINT "Quantity is:
"D%(J,0)
630 PRINT:INPUT "Alter to ";D:IF D>9999
THEN 630
```

```
640 D%(J,0)=D
650 GOTO 580
660 CLS:GOSUB 1270
670 FOR J=1 TO N
680 IF A=4 AND D%(J,0)>D%(J,1) THEN 750
690 IF A<>5 THEN 710
700 YR=VAL(RIGHT$(D$(J,2),2)):IF YR>Y OR
 Y=YR AND M<VAL(MID$(D$(J,2),3,2)) THEN
750
710 REM** OTHER SEARCHES **
720 REM** OTHER SEARCHES **
730 IF FP=1 THEN GOSUB 1340:GOTO 750
740 GOSUB 1290
750 NEXT
760 PRINT:PRINT:PRINT:PRINT" < SPACE BAR
 = CONTINUE : P = PRINT >"
770 A$=UPPER$(INKEY$):IF A$<>"P" AND A$<
>" "THEN 770
780 FP=0:IF A$="P" THEN GOSUB 1320:FP=1:
GOTO 670
790 GOTO 190
800 CLS:M=0:WHILE M<1 OR M>12
810 PRINT:INPUT "Which month (1 - 12)";M
$:M=VAL(M$):WEND
820 Y=-1:WHILE Y<0 OR Y>99
830 PRINT:INPUT "Which year (0 - 99)";Y$
:Y=VAL(Y$):WEND
840 GOTO 660
850 OPENOUT T$
860 PRINT #9,N
870 FOR J=1 TO N
880 FOR K=0 TO 4
890 PRINT #9,D$(J,K)
900 NEXT
910 FOR K=0 TO 2
920 PRINT #9,D%(J,K)
930 NEXT:NEXT
940 CLOSEOUT
950 PRINT:PRINT "Back-up? (Y/N)"
960 GOSUB 1000
970 IF A$="Y" THEN 850
980 PRINT:PRINT "Stock take finished"
990 END
1000 A$=UPPER$(INKEY$)
1010 IF A$<>"Y" AND A$<>"N" THEN 1000
```

```
1020 RETURN
1030 INPUT "CODE";D$:D$(J,0)=LEFT$(D$,4)
1040 INPUT "DESCRIPTION";D$:D$(J,1)=LEFT
$(D$,10)
1050 INPUT "DATE (DDMMYY)";D$:D$(J,2)=LE
FT$(D$,6)
1060 INPUT "QUANTITY";D:IF D>9999 THEN 1
060
1070 D%(J,O)=D
1080 INPUT "MINIMUM STOCK";D:IF D>9999 T
HEN 1080
1090 D%(J,1)=D
1100 INPUT "RE-ORDER QUANTITY";D:IF D>99
99 THEN 1100
1110 D%(J,2)=D
1120 INPUT "FIELD 1";D$:D$(J,3)=LEFT$(D$
,4)
1130 INPUT "FIELD 2";D$:D$(J,4)=LEFT$(D$
,4)
1140 CLS:PRINT "Data to be entered is:"
1150 PRINT:PRINT "CODE:"TAB(20)D$(J,0)
1160 PRINT "DESCRIPTION:"TAB(20)D$(J,1)
1170 PRINT "DATE:"TAB(20)D$(J,2)
1180 PRINT "QUANTITY:"TAB(19)D%(J,0)
1190 PRINT "MINIMUM STOCK:"TAB(19)D%(J,1
)
1200 PRINT "RE-ORDER QUANTITY:"TAB(19)D%
(J,2)
1210 PRINT "FIELD 1:"TAB(20)D$(J,3)
1220 PRINT "FIELD 2:"TAB(20)D$(J,4)
1230 PRINT:PRINT:PRINT:PRINT;TAB(20) "Al
l OK? Y/N"
1240 GOSUB 1000
1250 IF A$="N" THEN CLS:GOTO 1030
1260 RETURN
1270 PRINT "CODE"CHR$(24) " DESCRIPT "CH
R$(24) " DATE "CHR$(24) " QY "CHR$(24) "
 MN "CHR$(24) " RE "CHR$(24) " F1 "CHR$(
24)" F2 "CHR
$(24)CHR$(8)
1280 RETURN
1290 PRINT LEFT$(D$(J,0)+"    ",4)CHR$(2
4)LEFT$(D$(J,1)+"          ",10)CHR$(24)
D$(J,2)CHR$(24)RIGHT$("    "+STR$(D%(J,0
)),4)CHR$(24
)RIGHT$("    "+STR$(D%(J,1)),4);
1300 PRINT CHR$(24)RIGHT$("    "+STR$(D%
(J,2)),4)CHR$(24)LEFT$(D$(J,3)+"    ",4)
CHR$(24)LEFT$(D$(J,4)+"    ",4)CHR$(24)C
HR$(11)
1310 RETURN
1320 PRINT #8, "CODE"TAB(7)"DESCRIPTION"
TAB(20)"DATE"TAB(28)"QUANTITY"TAB(38)"MI
N.QTY"TAB(48)"RE-ORDER"TAB(59)"FIELD 1"T
AB(69)"FIELD
 2"
1330 RETURN
1340 PRINT #8,D$(J,0)TAB(7)D$(J,1)TAB(20
)D$(J,2)TAB(28)RIGHT$("    "+STR$(D%(J,0
)),4)TAB(38)RIGHT$("    "+STR$(D%(J,1)),
4)TAB(48)RIG
HT$("    "+STR$(D%(J,2)),4)TAB(59)D$(J,3
)TAB(69)D$(J,4)
1350 RETURN
1360 PRINT TAB(5)" < SPACE BAR TO CONTIN
UE >"
1370 A$=INKEY$:IF A$<>" " GOTO 1370
1380 RETURN
```

Variations

One of the features of this program is that it is adaptable. The two spare items, Field 1 and Field 2 can be used to hold any kind of data. If you have decided on a definite use for these columns, change their headings by amending lines 1120, 1130, 1210, 1220, and 1320.

The Old List normally requires you to enter a month-and-year date. To change to a day-and-month date, make these changes in the program:

Alter the message in line 810 to "Which day (1–31) "

Alter line 800 to:

```
800 CLS : M = 0: WHILE M<1 OR M >32
```

Alter the message in line 830 to "Which month (1–12) "

Alter line 820 to:

```
820 Y = 0 : WHILE Y<1 OR Y>12
```

Change line 700 to:

```
700 YR=VAL(MID$(D$(J,2),3,2)):IF YR>Y OR Y=YR
AND M <VAL(LEFT$(J,2),2) THEN 750
```

In Options 3 to 5, the program searches the stock list looking for records having a certain feature. Examples are the date (Old List) and the relation between stock quantity and minimum re-order quantity (Order List). Lines 680 and 700 identify such records, and make the computer skip over any which do not conform to the specification. Space has been left in the program at lines 710 and 720 for you to write your own search routines. These could be used in conjunction with information in any column, including Fields 1 and 2. You would need to add an extra option to the menu, to allow the requirements to be keyed in. Then the program would go to line 660 for the search to be made and the list displayed. In this way you could make the computer display all records which refer to the same supplier, for example, or prepare a list of all items falling within a given price range. With suitable coding in Fields 1 and 2, the program could display a list of foodstuffs needing special storage requirements, or those which are to be purchased in large quantities in certain seasons, for deep-freezing. If you do not wish to go to the trouble of writing extra routines, it may be possible to make use of the code in the first column of the list. If the code is used to indicate certain categories of item, Option 3 will allow you to find all of these, displaying them one at a time.

4
Pie Chart

A very effective way to display information is to use a pie chart. Most readers will already be very familiar with this form of presentation. A circular area is divided into a number of segments or slices, just as you would divide a real pie when sharing it among the family. The slices of a pie chart are usually of different colours and each is named to show what it represents. Pie charts are the ideal way of representing sharing. If you are shown a pie chart dealing with the expenditure of your local council, for example, you can instantly see what share of it is spent on each of the various services. Many people find it hard to take in the meaning of tables of figures but anyone can clearly see from a pie chart that the council is spending a third of its budget on education, for example, a quarter on leisure services, while it spends only a twentieth on street cleaning. We do not really want to know the exact figures – but just the way the money is shared. Then we can ascertain that each service is obtaining what we think is its fair share of the money.

Pie charts are useful in many other fields. Use them to display geographical, economic, demographic, or many other kinds of numeric data. Use them to demonstrate information relating to sales, production and company finances. Use them to analyse your daily diet, showing what shares of it are made up by carbohydrates, fats, proteins and fibre.

Using the program

The program gives you the opportunity to save details of pie charts on tape, so that you can have one or more sets of them ready for a display or demonstration. The first question that the computer asks is 'Keying in new chart? OR Loading from tape?'. If you are keying in new charts, key 'N' (for 'new'). If you are loading from tape, key 'T' (for 'tape') and follow the instructions given in the next section but one, *Loading from tape*.

If you key 'N' you will then be asked 'How many charts?'. Key in a number between 1 and 20, then press ENTER. You are now asked to key in the details of each of the charts, one chart at a time. You are first asked its name. Key in a name by which you can identify the chart later. Preferably, it should have less than seven letters. Next, you are asked 'How many slices?' A pie cannot be cut into fewer than two slices, and in this program the greatest number of slices allowed is fifteen. If a chart has more than fifteen slices, many of the thinner ones are displayed rather poorly. In any event, a chart with lots of small slices is relatively difficult to interpret – one might just as well be looking at a table of figures. The essence of a pie chart is that it is boldly drawn with most of the slices fairly large. If your data has more than fifteen headings, decide at this stage which items can be added together and displayed under one heading.

You will then be asked to key in the data for each slice.

The two questions asked are 'Label?' and 'Value?'. For the Label, key in the heading itself (education, protein, exports, etc.). For the Value, it does not matter what units the values are expressed in, provided that all slices of the same chart have the same units. For example, if the figure for a slice is '12' and this represents twelve thousand pounds, then all figures entered for the other slices of that pie chart must represent thousands of pounds. Similarly, if the figure you enter for the protein in your diet is the number of grams you eat in a day, then all figures for that chart must be 'numbers of grams per day'. You cannot have different units in the same chart. For example, you cannot show in one chart the numbers of grams of food materials and the number of calories they contain.

If your data is available as percentages, you can enter these as values. In fact, the program converts *all* kinds of data into percentages when it is working out how to plot the charts. Any slices that occupy less than 1% of the pie are not displayed on the chart.

After you have keyed in all the Labels and Values for a chart, you are asked 'ALL OK?'. If you have entered something wrongly, key 'N' and you can then enter everything again.

If you key 'Y', key in the details of the next chart, repeating this process until details of all charts have been entered.

Viewing the charts

The screen then clears and the first chart is displayed. Its name and number appear at the top of the screen. The chart is plotted in the centre of the screen, in up to fifteen colours. When the plotting of the pie is completed, the Labels and Values are displayed in a table below the chart. The Labels and Values are displayed in the same colours as the slices, so that it is easy to relate the Label and Value to the appropriate slice. The list of Labels and Values scrolls slowly up the lower part of the screen. Only five are on view at any one instant, but the whole list is repeatedly displayed. The segments are drawn in clockwise order, starting at the top, and the Labels and Values are always displayed in order in two columns, so it is easy to see which label refers to which slice.

When the plotting is complete, the computer waits for you to examine the chart. When you have finished, press the space-bar. The screen clears and the next chart is plotted. When all charts have been examined, pressing the space-bar returns you to the first chart again. You can run round the whole set of charts as many times as you require.

Pressing 'M' or 'S' takes you to one of two other routines. 'M' stands for 'more'. This key takes you to a routine which allows you to add more charts. You will see the message 'How many charts?' The number referred to is the *total* number of charts. If you have already keyed in three charts, for example, you will now be invited to increase that number to anything between four and twenty. If you want to add just one more chart, making the total up to four, key '4'. You are then asked to enter details of the new chart or charts, in turn. The routine for this is exactly the same as before. After this is complete, the extended set of charts may be viewed, one after the other. As before, it may be repeated as many times as you wish.

If you want to save the charts to tape, key 'S' after viewing any chart. The first question is 'File name?'. Key in a name by which to identify the set of charts. Then press ENTER. The message 'Press REC and PLAY then any key' appears. You can now rewind or fast-forward the tape to its correct position, ready for saving the chart details. After this, press RECORD and PLAY on the recorder, then press any key on the micro. Saving takes only about half a minute, less if you have few charts. You are then asked if you wish for a backup copy. If so, key 'Y' and repeat the procedure described above. If not, key 'N', and the program ends.

Loading from tape

If you already have chart details saved on tape and want to view these charts, or perhaps add other charts to the set, place the tape in the recorder. After you have keyed 'T', as described earlier, you are asked 'File name?'. Type in the file name for the set of charts, then press ENTER. You can now rewind or fast-forward the tape to find the place where the file is recorded. Then press the PLAY button on the recorder then any key. In a few tens of seconds, loading will be complete.

As soon as loading is finished, the screen clears and the first chart is automatically displayed. You can then view this and any other of the charts, add more charts to the set, if you wish or save the extended set, as described in the section *Viewing the charts*, above.

Keying in

The variable in lines 720 and 740 is PI, the micro's built-in variable having the value 3.14159265. Do not key 'P1' by mistake.

Program design

20–110 Initialising.
120–190 New charts or tape?
200 How many charts?
210–420 Collecting details of charts.
430–520 Calculating percentages for each slice of each chart.
530–640 Loading from tape.
650–840 Displaying charts.
850 To key in more charts.
860–1010 Saving to tape.
1020–1060 Subroutine to find out how many charts.

The program

```
10 REM ** PIE CHART **
20 MODE 0
30 OPENOUT "AMSTRAD"
40 MEMORY HIMEM-1
```

```
50 CLOSEOUT
60 DEFINT J,M,N,P,S,X,Y
70 WINDOW #1,1,20,21,25:ORIGIN 0,0,0,639
,81,399
80 ORIGIN 320,240
90 INK 5,21:INK 9,18:INK 12,12:INK 14,15
:INK 15,5
100 DIM VS(20,15),LS$(20,15),NA$(20),PS%
(20,15),S%(20)
110 SJ=1:P=0:R=160
120 CLS:LOCATE 5,1:PRINT "PIE CHART":LOC
ATE 5,2:PRINT "_________"
130 A$="":WHILE A$<>"N" AND A$<>"T"
140 LOCATE 1,8:PRINT "Keying in new char
t?":LOCATE 1,10:PRINT "OR"
150 LOCATE 1,12:PRINT"Loading from tape?
160 LOCATE 5,16:PRINT "(N/T)?"
170 A$=UPPER$(INKEY$)
180 WEND
190 IF A$="T" THEN 530
200 GOSUB 1020
210 J=SJ:WHILE J<>P+1
220 A$="":WHILE A$<>"Y"
230 CLS:PRINT"PIE CHART NO."J
240 PRINT:INPUT "Chart name ";NA$(J)
250 NA$(J)=LEFT$(NA$(J),6)
260 S(J)=0:WHILE S(J)<1 OR S(J)>15
270 LOCATE 1,5:PRINT"How many slices":IN
PUT "(2-15)";AS$
280 IF AS$="" THEN 270
290 S(J)=VAL(AS$):
300 WEND
310 FOR K=1 TO S(J)
320 LOCATE 1,K*2+6:INPUT "Label";LS$
330 LS$(J,K)=LEFT$(LS$,10)
340 VS(J,K)=0:WHILE VS(J,K)=0
350 LOCATE 10,K*2+7:INPUT "Value";VS(J,K
)
360 WEND
370 NEXT
380 PRINT:PRINT "ALL OK? (Y/N)";
390 A$=UPPER$(INKEY$):IF A$="" THEN 390
400 WEND
410 J=J+1
420 WEND
```

```
430 FOR J=1 TO P+1
440 TP=0:TV=0:MP=0:PS(J,0)=100
450 FOR K=1 TO S(J):TV=TV+VS(J,K):NEXT
460 FOR K=1 TO S(J):PS(J,K)=INT(VS(J,K)/
TV*100+0.5)
470 TP=TP+PS(J,K)
480 IF PS(J,K)>MP THEN MP=PS(J,K):M=K
490 NEXT
500 PS(J,M)=PS(J,M)+100-TP
510 NEXT
520 GOTO 650
530 PRINT:INPUT "Filename";NF$
540 OPENIN NF$
550 INPUT #9,P
560 FOR J=1 TO P
570 INPUT #9,NA$(J)
580 INPUT #9,S(J)
590 FOR K=1 TO S(J)
600 INPUT #9,VS(J,K)
610 INPUT #9,PS(J,K)
620 INPUT #9,LS$(J,K)
630 NEXT:NEXT
640 CLOSEIN
650 J=1
660 A$=" ":WHILE A$=" "
670 CLS:IF J=P+1 THEN J=1
680 CLS:LOCATE 1,1:PRINT NA$(J):LOCATE 1
8,1:PRINT J;
690 A=0
700 FOR K=1 TO S(J)
710 FOR L=1 TO PS(J,K)
720 A=A+PI/100:X=R*SIN(A):Y=R*COS(A)
730 DRAW X,Y,K
740 A=A+PI/100:X=R*SIN(A):Y=R*COS(A)
750 DRAW X,Y,K:DRAW 0,0,K
760 NEXT:NEXT
770 J=J+1:K=0
780 K=K+1:IF K>S(J-1) THEN K=1
790 PEN #1,K
800 PRINT #1,LS$(J-1,K)"       "VS(J-1,K)

810 FOR L=1 TO 1000:NEXT
820 A$=UPPER$(INKEY$):IF A$="" THEN 780
830 PEN 1
840 WEND
```

```
850 IF A$="M" THEN GOSUB 1020:GOTO 210
860 CLS:INPUT "File name";NF$
870 OPENOUT NF$
880 PRINT #9,P
890 FOR J=1 TO P
900 PRINT #9,NA$(J)
910 PRINT #9,S(J)
920 FOR K=1 TO S(J)
930 PRINT #9,VS(J,K)
940 PRINT #9,PS(J,K)
950 PRINT #9,LS$(J,K)
960 NEXT:NEXT
970 CLOSEOUT
980 PRINT:INPUT "Backup";A$
990 IF UPPER$(A$)="Y" THEN 870
1000 PRINT:PRINT "Pie Chart finished":PR
INT
1010 END
1020 WHILE VAL(A$)<=P OR VAL(A$)>20
1030 CLS:PRINT "How many charts?":LOCATE
 5,3:PRINT "("P+1;:INPUT "- 20)";A$
1040 WEND
1050 SJ=P+1:P=VAL(A$)
1060 RETURN
```

5
Index

This program lets you enter items in any order and then displays them in alphabetical order. In short, it produces an index. The program can help you keep a list of your tape collection, a list of members of your club, or a list of names and addresses of your friends. Authors will find it very helpful when indexing a book, and you can also use it for producing catalogues, glossaries, and any other kind of list in which the items are to be arranged in alphabetic order.

The indexed list may be saved to tape and added to at some future date. There is no need to enter all the items at one session. If you have a printer attached to your micro, you can print out the whole or any section of the list. The list can contain up to 800 items, provided that your entries are not excessively long. The program has a special feature to allow you to see how much memory is left.

Using the program

The first question the computer asks is 'Load from tape?'. If you wish to look at or add to a list that you have already saved on tape, key 'Y'. If you are starting a new list, key 'N', and you will be taken directly to the part of the program which deals with entering items (see later).

On keying 'Y' you are asked 'Name of file?'. Key in the file name that you used for the list you want to load, then press ENTER. You will now be able to fast-forward or rewind the tape to locate the required part of the tape. Press PLAY on the recorder then any key. After several seconds – or up to five minutes if the list is nearly full – the screen clears and the query 'Entry?' is displayed.

Making entries

The query 'Entry?' is asking you to type in the key-word or key-words for the entry. This is the word (or words) which are to be sorted alphabetically.

At the bottom right corner of the screen, you will see 'Entry no.'. This is the number to be given to the next entry to be put on the list. If you are starting a new list, the entry number is 1. If you have just loaded in a list from tape, this number is 1 greater than the number of items already on the list. If the number is close to 800, the list is almost full. The computer will 'beep' at you as you make entries 795 to 800 to warn you of this. Below the entry no. is the statement 'Memory ='. This tells you how many bytes of memory are available for holding entries. If you are starting a new list, this number is quite large (about 27000). As you enter more and more items, the number decreases. Sometimes it decreases a little, sometimes a lot, while at other times it stays the same. At every 20th entry it *increases* by a large amount, when the computer rearranges its store of data to make it more compact. The general trend is downward and the computer beeps at each entry when the number falls below 200. You may be able to squeeze in a few short entries at that stage, but do not try to type in long ones. If in doubt, save your existing entries before trying to cram in a few more.

Before continuing with a description of the program, the meaning of 'alphabetical order' needs to be clarified. The program works according to several rules:

(1) It sorts on a letter-by-letter basis, not on a word-by-word basis. Here are items sorted in the two different ways:

Letter-by-letter	*Word-by-word*
abacus	A.C. mains
A.C. mains	A.C. theory
Acorn computers	A.M. radio
A.C. theory	abacus
A.M. radio	Acorn computers
Arcade games	arc-back
arc-back	Arcade games

Letter-by-letter sorting ignores all punctuation and spaces. Word-by-word sorting sorts by the first word (or the first single letter) of an item, ignoring subsequent words or letters. Then it sorts the second words (or second single letters), and so on. Parts of hyphenated words are usually taken to be separate words.

(2) All punctuation and spaces are ignored in sorting.

(3) Whether letters are capitals or lower-case letters makes no difference to the sorting.

(4) Names that begin with 'St.' are sorted as if they begin with 'Saint'.

(5) Names that begin with 'M' ', 'Mc', or 'MC' are sorted as if they begin with 'Mac'.

Note that although the program may ignore punctuation, or may sort 'Mc' as 'Mac', all entries are displayed exactly as you type them in. Probably you will wish to key in entries in upper-case (capitals) and lower-case, just as on a typewriter.

It is important that only the part of the item that is to be used for alphabetic sorting is keyed in as the 'Entry'. When this has been typed in, press ENTER. Lower down on the screen you will then see the word 'Reference?'. This is asking you to key in the remainder of the item, if any. It is here that you key in page numbers, if you are indexing a book, for example. You can key in a single page number, a span of several pages (e.g. '123-7') or a cross-reference (e.g. (see computers, Amstrad)). If you are preparing an address book, the address and telephone number can be typed in as 'Reference'. The only point to remember is that the program will not hold as many entries if you type a lot into 'Reference'.

As soon as you have completed the reference section, press ENTER. The screen clears and you are asked for the next 'Entry'. If you are indexing a book, you will sometimes need to add further page numbers to an existing entry. This is done by typing in the entry again. Make sure it is spelt exactly as before, with the same punctuation and spaces, if any. The new page number may then be typed in as the 'Reference'. The computer adds this new page number to those already listed under this item. You will see that the 'Entry no' at the bottom of the screen does not increase after such an operation. No new entry has been created. You are merely adding a page number to the existing entry. Doing this requires very little additional memory, usually none at all.

Other commands

The 'Entry?' message offers you the opportunity to make the program do other things beside accept entries. Instead of typing an entry, key in one of the following:

(1) **=** The computer displays the list of entries, with their references. If you type '=' alone, you will be shown the whole list. Press ESC to halt the display; to see the next screenful, press the space-bar. If you type '=' followed by one or more letters, you will see just that part of the list which contains words beginning with that letter (or letters). For example, typing '=D' (no space between '=' and 'D') gives you all entries beginning with 'D' or 'd', arranged in alphabetical order. Typing '=MIC' would give you all words beginning with these three letters, such as 'mice', 'Michaelmas' and 'microcomputer'. If there are no words beginning with these letters, or when the computer has displayed all that there are, you see the message 'Space Bar to continue: 'R' to review'. Pressing the space-bar takes you back to 'Entry?', while pressing 'R' lets you see the whole or the specified part of the list again.

(2) ¦ (This appears on the screen as |). It is used for deleting items from the list. If you type ' ¦ ' on its own, you will be able to look through the whole list, as when you type '='. If you type ' ¦ ' followed by one or more letters, you will be shown only those words beginning with the letters you have typed. Each item is displayed separately and you are asked to key 'D' if you want it to be deleted, or 'R' if you want it to be retained (*not* deleted). Take care, for an item once deleted cannot be recovered; you will have to type it in again! Deleting an item does not free the memory it occupied. Deleted items are still included in the maximum of 800 items that the program allows.

When all the items called for have been displayed the message 'Deletions completed' appears. After a few seconds delay, you are taken back for the next entry.

(3) { (Press SHIFT and [ ). This takes you to the routine for saving the list on tape. If you have previously loaded the list from tape, the computer now reminds you what file name you used. It then asks you to key in the file name under which the list is to be saved. Probably you will choose to key in the same name, but you can use a different name if you wish. After you have keyed in the name, the message 'Press REC and PLAY then any key' appears. Rewind the tape, if necessary, so that it is ready to receive the recording. Press REC and PLAY on the recorder, then press any key on the micro. Saving takes up to five minutes if you have a full list. When saving is complete, the message 'BACKUP?' appears. If you want to take a backup copy,

key 'Y' and repeat the procedure described above. If not, key 'N' at which you will see the message 'Index program finished'.

(4) ↑ (On the same key as '£'.) This has the same effect as '=' except that the list or part of the list is printed, instead of being displayed on the screen.

Keying in

Note the space in "St. " used to identify a saint's name. Take particular care with typing lines 580 to 640 and 740 to 870. The ON ERROR command is in force in these sections of the program. If you have made a typing error you will not get the usual error messages to tell you what you have done wrong. If you find that the displaying of the list, the printing of the list or the deletion routine always seem to end too quickly it could be that there are errors in typing these lines.

Program design

20–80 Initialising arrays and variables.
90–120 Load from tape?
130–210 Loading from tape.
220–270 'Entry?' display.
280–330 Inputting the entry.
340–370 To various routines.
380 Rejecting entry when memory is full.
390 Inputting reference.
400 To convert to upper-case etc.
410–430 Dealing with 'Mac's and 'St. 's.
440–550 Sorting by calculating key pointers.
560–700 Displaying or printing the list.
710–900 Deleting routine.
910–1050 Saving to tape.
1060–1110 Subroutine to convert an entry into a string of capital letters with all spaces and punctuation omitted, ready for sorting.
1120–1130 Subroutine to produce a delay.

Points of interest

Large amounts of memory can be wasted if strings are frequently moved around in memory. But how can we avoid doing this when an index requires that the entries be sorted into alphabetical order? How can we avoid rearranging the entries every time a new entry is made?

INDEX does not attempt to move the entries around. Each entry goes into a new location in the array E$() as it is keyed in, and stays there permanently. At the same time the converted 'all capitals' version, used in sorting, goes to the corresponding location in F$(). The reference goes into R$(). Thus, a given entry always has the same location number in all three arrays. The other array used in this program is K(), which holds what are called the *key pointers*. These are numbers which 'point to' locations in F$() (or E$() and R$()). The computer finds its way to the first item in alphabetical order by using the *head-of-list pointer*, H. If H has the value 25, for example, it means that the alphabetic first entry is number 25, which is to be found at E$(25), and F$(25), with its reference at R$(25). When asked to display the whole list, the computer first prints what it finds at E$(25) and R$(25). Then it looks at the number stored in K(25). This tells it where the next entry in alphabetic order is to be found. K(25) might hold the value 12, for example, indicating that the second item on the list is at E$(12) and R$(12). Then, at K(12) it finds the pointer to the third item, and so on to the end of the list. When it reaches the last item in the list, it finds in K() the value −1, indicating that it has reached the end. The computer does not allow such expressions as K(−1), and at this stage the ON ERROR condition causes the computer to go to the end of the routine.

Using this system, it is necessary only to alter a few values in K() whenever an item is to be added to or deleted from the list. Changing values in a numeric array requires no additional memory. This technique saves much time too. If we were to keep entries in E$(), F$() in alphabetical order we would need, on average, to shift half of them to new locations every time a new entry was made. Changing the few relevant key pointers is very much quicker. A list of items such as this is sometimes referred to as a *linked list*.

The program

```
10 REM ** INDEX **
20 MODE 1
30 OPENOUT "AMSTRAD"
40 MEMORY HIMEM-1
50 CLOSEOUT
60 DEFINT C,H-N,P,Q
70 DIM E$(801),F$(801),K(800),R$(800)
80 H=1:K(1)=-1:PC=1:N=1:X$=""
90 LOCATE 1,10:PRINT "Load from tape? (Y
/N) ";
100 A$=UPPER$(INKEY$)
110 IF A$<>"Y" AND A$<>"N" THEN 100
120 IF A$="N" THEN 220
130 PRINT:PRINT:INPUT "Name of file";X$

140 OPENIN X$
150 INPUT #9,N
160 INPUT #9,H
170 FOR J=1 TO N:INPUT #9,E$(J):NEXT
180 FOR J=1 TO N:INPUT #9,F$(J):NEXT
190 FOR J=1 TO N:INPUT #9,K(J):NEXT
200 FOR J=1 TO N:INPUT #9,R$(J):NEXT
210 CLOSEIN
220 CLS:IF N=801 THEN LOCATE 24,4:PRINT
"** Index FULL **";CHR$(7)
230 LOCATE 24,21:PRINT "Entry no: "N
240 IF N/20=INT(N/20) THEN F=FRE("")
250 LOCATE 24,22:PRINT "Memory = "FRE(0)
260 IF N>795 OR FRE(0)<200 THEN PRINT CH
R$(7)
270 LOCATE 2,3:PRINT "Entry ?":PRINT CHR
$(10):PRINT CHR$(10)
280 E$(N)="":Y$=""
290 WHILE Y$<>CHR$(13)
300 Y$=INKEY$:IF Y$="" THEN 300
310 IF Y$=CHR$(127) THEN E$(N)=LEFT$(E$(
N),LEN(E$(N))-1):PRINT Y$;:GOTO 330
320 IF Y$<>CHR$(13) THEN PRINT Y$;:E$(N)
=E$(N)+Y$
330 WEND
340 IF LEFT$(E$(N),1)="^" THEN P=8
350 IF LEFT$(E$(N),1)="=" OR P=8 THEN 56
0
```

```
360 IF LEFT$(E$(N),1)="¦" THEN 710
370 IF LEFT$(E$(N),1)="{" THEN 910
380 IF N>800 OR FRE(0)<100 THEN 220
390 LOCATE 1,12:INPUT "Reference";R$
400 GOSUB 1060
410 T$=LEFT$(E$(N),2):IF T$="M'" THEN F$
(N)="MAC"+MID$(F$(N),2)
420 IF T$="MC" OR T$="Mc" THEN F$(N)="MA
C"+MID$(F$(N),3)
430 IF LEFT$(E$(N),4)="St.  " THEN F$(N%
)="SAINT"+MID$(F$(N),3)
440 IF N=1 THEN 540
450 L=0:C=H
460 D$=F$(C):QC=PC:PC=C:C=K(C)
470 IF C=-1 AND F$(N)>D$ THEN K(N)=-1:K(
PC)=N:L=1:GOTO 490
480 IF F$(N)>D$ THEN 460
490 IF L=1 THEN 540
500 IF F$(N)=D$ THEN R$(PC)=R$(PC)+","+R
$:GOTO 220
510 IF PC=H THEN H=N:K(N)=PC:GOTO 540
520 K(QC)=N
530 K(N)=PC
540 R$(N)=R$:N=N+1
550 GOTO 220
560 CLS:IF H=-1 THEN LOCATE 1,4:PRINT "I
ndex empty":GOSUB 1120:GOTO 220
570 GOSUB 1060
580 ON ERROR GOTO 640
590 C=H
600 IF F$(C)<F$(N) THEN C=K(C):GOTO 600
610 I=INSTR(F$(C),F$(N))
620 IF I=1 AND P=0 THEN PRINT E$(C)"
"R$(C):C=K(C):GOTO 610
630 IF I=1 AND P=8 THEN PRINT #8,E$(C)"
   "R$(C):C=K(C):GOTO 610
640 RESUME 650
650 P=0
660 PRINT:PRINT:PRINT " Space bar to con
tinue: 'R' to review"
670 A$=UPPER$(INKEY$)
680 IF A$<>" " AND A$<>"R" THEN 670
690 IF A$="R" THEN CLS:GOTO 580
700 GOTO 220
710 CLS:IF H=-1 THEN LOCATE 1,4:PRINT"In
```

```
dex empty":GOTO 890
720 PRINT"D = Delete : R = Retain"
730 GOSUB 1060
740 ON ERROR GOTO 870
750 C=H
760 IF F$(C)<F$(N) THEN QC=PC:PC=C:C=K(C
):GOTO 760
770 LOCATE 1,3
780 I=INSTR(F$(C),F$(N))
790 IF I=1 THEN PRINT E$(C)"  "R$(C)"  ?
 ";:QC=PC:PC=C:C=K(C)
800 IF I=0 THEN 870
810 A$=UPPER$(INKEY$):IF A$<>"D" AND A$<
>"R" THEN 810
820 PRINT A$
830 IF A$="R" THEN 780
840 IF PC=H THEN H=C:GOTO 780
850 IF PC<>H THEN K(QC)=K(PC):PC=QC
860 GOTO 780
870 RESUME 880
880 PRINT:PRINT "Deletions completed"

890 GOSUB 1120
900 GOTO 220
910 CLS:IF X$<>"" THEN PRINT:PRINT:PRINT
 "This was saved as file '";X$;"'"
920 PRINT:PRINT:INPUT "What file name";X
$
930 OPENOUT X$
940 PRINT #9,N
950 PRINT #9,H
960 FOR J=1 TO N:PRINT #9,E$(J):NEXT
970 FOR J=1 TO N:PRINT #9,F$(J):NEXT
980 FOR J=1 TO N:PRINT #9,K(J):NEXT
990 FOR J=1 TO N:PRINT #9,R$(J):NEXT
1000 CLOSEOUT
1010 PRINT:PRINT:PRINT "Backup? (Y/N) ";
1020 A$=UPPER$(INKEY$):IF A$<>"Y" AND A$
<>"N" THEN 1020
1030 IF A$="Y" THEN 910
1040 PRINT:PRINT:PRINT "Index program fi
nished"
1050 END
1060 F$(N)=""
1070 FOR J=1 TO LEN(E$(N))
```

```
1080 M$=UPPER$(MID$(E$(N),J,1)):M=ASC(M$
)
1090 IF M>64 AND M<91 THEN F$(N)=F$(N)+C
HR$(M)
1100 NEXT
1110 RETURN
1120 FOR J=1 TO 500:NEXT
1130 RETURN
```

6

Compare It

Mr Smith grows two varieties of tomato in his greenhouse. He has the same number of plants of each variety and looks after them in the same way. Each season he keeps a record of how many kilograms of tomatoes are produced by all of the plants of each variety. After six seasons he looks at his figures:

Variety	*Seasonal production*					
Catsup	100	120	150	130	160	135
Ketchup	140	110	180	150	165	125

The total production of Catsup is 795 kg, while that of Ketchup is 870 kg. There *is* a difference – but this is to be expected. After all, it would be very unusual if both varieties gave *exactly* the same total. The figures show that some years give better yields for both varieties, some give worse for both. Although Ketchup has the higher total, there are two years in which it does worse than Catsup. The situation is far from clear-cut. Mr Smith wants to know if Ketchup is a higher-yielding variety than Catsup in the long run.

If only figures would give a straightforward answer to our queries! More often than not, there are uncertainties, as in the story above. The difference between the yields of the two varieties, totalled over the six years, *might* be due to Ketchup being the better-yielding variety overall. But it is also possible that the difference is merely the result of chance variations, of the kind that are obviously occurring with both varieties from year to year. What this program does is to help you compare two sets of data to see if the difference between them is more likely to be real, or more likely to be due to chance. The program tells you how significant the difference is.

COMPARE IT is not limited to finding out about tomatoes. In almost any instance in which there are two sets of data to be compared, this program can be brought into use. It is interesting, and possibly profitable, to use it to compare the weights of sweets, potato crisps, etc., found in packets sold by two different manufacturers. Some firms put more into their packets, on average, than do other firms, even though the quantity marked on the packet may be exactly the same for both firms. Weighing the contents of several packets from each of two firms can show you which is the better buy. Use this test to find out if the difference between the quantities packeted by each firm really is significant. The program is able to accept several different kinds of data. It accepts integer values, such as those Mr Smith obtained by weighing his tomatoes. It can also accept numbers with decimal places, and negative numbers. You might measure the minimum temperature in two different parts of your garden during the winter. This would give you two sets of temperatures, which would almost certainly contain some negative (below freezing-point) values. Comparing the two sets would tell you whether one part of the garden was warmer than the other, on average. If the difference turned out to be significant, the warmer spot would be the place to site a greenhouse, a rabbit hutch, or a bee-hive, or perhaps to plant a tender shrub.

The usefulness of the program is widened by the fact that it works just as well with numbers which represent ranks, grades, or ratings as it does with ordinary numbers. If you are trying to find out which of two makes of jam have the best taste, it is not possible to rate the taste in grams, centimetres or any other recognised unit. Instead, you would ask, say, twenty different people to taste samples of the jam. You would ask them to grade the two jams on a scale running from 1 (poor), through 5 (average) to 10 (excellent). The sets of grades for each kind of jam are compared, using the program.

Grading is used in many kinds of application where quantitative measurements are difficult or impossible:

Tasting of foods, teas, wine.
Assessing visual features, such as the colours of flowers.
Assessing other features such as the texture of fabrics.
Rating human performance, such as musical and sporting skills.
Rating human reactions, such as satisfaction with a certain product or service, happiness in a given job, friendliness with others.

There seems to be no end to the applications of COMPARE IT!

If you bear a few special points in mind when using the program, you will be more likely to get a clear answer from the test. There must be at least five items of data in each of the two sets to be compared. There need not be the same number of items in each set; for example, if you wanted to conduct a popularity survey of two TV programmes, you could ask fifteen passers-by to give their rating of a particular TV programme, yet interview twenty other passers-by about the other programme. But the more nearly equal the sizes of the two groups, the more reliable will be the result of the test.

Using the program

All you have to do is to key in the data and then wait a few tens of seconds for the micro to work out the results.

First of all you are asked to give a name to each of the two sets of data. For example, Mr Smith would have typed in 'CATSUP' and 'KETCHUP'. Then you are asked to key in the items of one set of data. Key them in one after another, pressing ENTER after you have entered each item. The program allows you to key in as many as fifty items in each set. When you have keyed in all the items of one set, key in '9999' as the final item. This figure will not be included among the data. It is just a convenient way of letting the micro know when the last item has been typed in.

Then you are asked to type in the data for the second set, finishing with '9999' in the same way.

After you have finished entering each set, you are asked 'OK? Y/N'. If you have many items of data, press ESC to view a screenful at a time. Press the space-bar to view the next screenful. Pressing 'N' at the end gives you a chance to start again if you have made a mistake in the keying. When you key 'Y' after the second set of data, the screen clears and a message is displayed reminding you that some time may be needed for the calculations to be completed. If you have only, say, ten or twenty items in each set, only a few tens of seconds will be required, but it could take a minute or two with larger amounts of data.

The final display gives you the computer's assessment of how much meaning you may safely attach to your data. Mr Smith would have been given the following assessment:

> 'THERE IS A CHANCE BETWEEN 60% AND 65% THAT THE CATSUP DATA DIFFERS FROM THE KETCHUP DATA'

This statement is explained as follows. The differences between the total six-year yields of the two varieties must be due to one of two things:

(1) One variety is a truly better-yielding variety than the other,

or

(2) There is no real difference between varieties; differences that exist are solely due to chance variations from plant to plant and from season to season.

From what Mr Smith saw on his computer screen he concluded that there is a 60% to 65% chance of the first explanation being the true one. Conversely, there is a 40% to 35% chance that the difference is purely random. Although there is a difference between the crops over the years, this difference is not significant when compared with the comparatively large variations in yield between different plants and from year to year. If the chance of there being a real difference is only 60% to 65%, it is not sensible to base any drastic decision on the yield figures. There is no real reason for giving up growing Catsup tomatoes, for example. If Mr Smith still wants to know if one variety is better than the other, he must carry on for a few more years and gather more data.

To sum up, the program tells us the likelihood of the difference between two sets of figures being significant. Most people would take a 90% chance (or more) to indicate a true difference, and take a 60% chance (or less) to mean that no significant difference exists.

Users of this program normally want to show that there is a difference between the *average* of values in one set of data and the *average* of values in the other. This is what Mr Smith wanted to do. It is also possible to have two sets of data in which the averages do not differ markedly, but in which the values of one set are spread over a much wider *range* than those of another. For example, two different brands of Cheddar cheese might, on average, be equally palatable. But one brand might be consistently average in palatability, while the other brand varies wildly. The worst cheeses of the second brand might be almost impossible to eat, while their best cheese might be superb. Given the results of tasting tests, this program would detect such a difference and assess the likelihood of its being significant.

The program detects difference of average values and difference of spread. It does not tell you which kind of difference (if any) occurs in your figures. Having obtained a result which declares that two sets of data are significantly different, inspect the original figures to see what kind of difference they show.

Keying in

Note the decimal point before the '5' in lines 230, 560 and 680. The crucial lines in the calculation are lines 660 to 680. Check these very carefully – especially the brackets. The upward pointing arrow in line 670 is the exponentiation sign. The key is marked '↑'. When typing the DATA statements of lines 920 to 940, note that they contain both *full-stops* (i.e. decimal points) and *commas*. These must all be typed exactly as printed in the book.

Program design

20–30 Initialising.
40–160 Collecting two sets of data.
170–180 Transferring data to arrays, ready for processing.
190 Warning of delay.
200–370 Sorting data into numerical order.
380–620 Resolving tied values.
630–650 Counting the number of runs.
660–680 Calculating the statistics.
690–880 Assessing probabilities and displaying the results.
890–910 Inviting another analysis.
920–940 Statistical look-up table.

Points of interest

The program uses quicksort, a fast sorting routine. It could be used in any other program in which numbers are to be sorted into ascending numerical order. The data is to be in an array S().

Note for the more statistically-minded reader: this program uses the Wald-Wolfowitz runs test, adapted for analysing two-sample data. Ties are resolved alternately one way or the other, any odd tie being resolved at random. The DATA statements contain the values of z associated with selected probability (one-tailed) levels.

The program

```
10 REM ** COMPARE IT **
20 MODE 1
30 DIM D(2,50),S(101),R(101),N$(2),N(2):
K=0:FT=0
40 CLS:INPUT "Name for first set of data
";N$(1)
50 PRINT:INPUT "Name for second set of d
ata";N$(2)
60 FOR J=1 TO 2
70 A$="N":WHILE A$="N":N(J)=1
80 D(J,K)=0:CLS:PRINT "Key in data for "
N$(J)
90 WHILE N(J)<50 AND D(J,N(J)-1)<>9999
100 PRINT:PRINT N(J);:INPUT"  ";D(J,N(J)
):N(J)=N(J)+1
110 WEND
120 CLS:PRINT "Data for "N$(J)" is:"
130 N(J)=N(J)-2:FOR K=1 TO N(J):PRINT:PR
INT D(J,K):NEXT
140 PRINT:PRINT:INPUT "All OK (Y/N) ";A$
:A$=UPPER$(A$)
150 WEND
160 NEXT
170 FOR J=1 TO N(1):S(J)=D(1,J):R(J)=0:N
EXT
180 FOR J=1 TO N(2):S(J+N(1))=D(2,J):R(J
+N(1))=1:NEXT
190 CLS:LOCATE 5,7:PRINT "Computing may
take a while"
200 M1=1:NT=N(1)+N(2)
210 SS(1,1)=1:SS(1,2)=NT
220 M2=SS(M1,1):M3=SS(M1,2):M1=M1-1
230 M4=M2:M5=M3:M6=S(INT(RND(1)*(M3-M2)+
0.5)+M2)
240 IF S(M4)<M6 THEN M4=M4+1:GOTO 240
250 IF M6<S(M5) THEN M5=M5-1:GOTO 250
260 IF M4>M5 THEN 280
270 M7=S(M4):S(M4)=S(M5):S(M5)=M7:M9=R(M
4):R(M4)=R(M5):R(M5)=M9:M4=M4+1:M5=M5-1
280 IF M4<=M5 THEN 240
290 IF M5-M2>=M3-M4 THEN 330
300 IF M4>=M3 THEN 320
```

```
310 M1=M1+1:SS(M1,1)=M4:SS(M1,2)=M3
320 M3=M5:GOTO 360
330 IF M2>=M5 THEN 350
340 M1=M1+1:SS(M1,1)=M2:SS(M1,2)=M5
350 M2=M4
360 IF M2<M3 THEN 230
370 IF M1>0 THEN 220
380 TS=1:TT=0:RS=0:RF=0
390 IF TS>NT THEN 550
400 IF S(TS)<>S(TS+1) THEN TS=TS+1:GOTO
390
410 TF=TS+1
420 IF TF>NT THEN 550
430 IF S(TF)=S(TF+1) THEN TF=TF+1:GOTO 4
20
440 NZ=0
450 FOR J=TS TO TF
460 IF R(J)=0 THEN NZ=NZ+1
470 NEXT
480 IF NZ=0 OR NZ=TF-TS+1 THEN TS=TF+1:G
OTO 390
490 RS=TS:RF=TF
500 FOR J=TS TO TS+NZ-1:R(J)=FT:NEXT
510 IF FT=0 THEN FT=1:GOTO 530
520 FT=0
530 FOR J=TS+NZ TO TF:R(J)=FT:NEXT
540 TS=TF+1:GOTO 390
550 IF FT=0 OR RS=0 THEN 620
560 IF RND(1)<0.5 THEN 620
570 NZ=0:FOR J=RS TO RF
580 IF R(J)=0 THEN NZ=NZ+1
590 NEXT
600 FOR J=RS TO RF-NZ:R(J)=1:NEXT
610 FOR J=RF-NZ+1 TO RF:R(J)=0:NEXT
620 R(NT+1)=R(NT)+1
630 U=0:FOR J=1 TO NT
640 IF R(J)<>R(J+1) THEN U=U+1
650 NEXT J
660 UU=2*N(1)*N(2)/(N(1)+N(2))+1
670 S=SQR(2*N(1)*N(2)*(2*N(1)*N(2)-N(1)-
N(2))/(N(1)+N(2))^2/(N(1)+N(2)-1))
680 Z=(ABS(UU-U)-0.5)/S
690 IF Z<0 THEN Z=0.001
700 IF N(1)*N(2)<18 OR N(1)<5 OR N(2)<5
THEN PRINT:PRINT "Not enough data":GOTO
```

```
890
710 R=0
720 R=R+1:READ ZL
730 IF ZL=Z THEN 750
740 IF ZL>Z THEN 720
750 IF R<10 THEN P=R/10
760 IF R>9 AND R<19 THEN P=R-9
770 IF R>18 THEN P=10+5*(R-19)
780 IF ZL<Z THEN 830
790 CLS:PRINT "There is a"100-P"% chance
 that the"
800 PRINT:PRINT N$(1)" data differs from
 the "
810 PRINT:PRINT N$(2)" data"
820 GOTO 890
830 IF R<11 THEN PL=(R-1)/10
840 IF R>10 AND R<20 THEN PL=R-10
850 IF R>19 THEN PL=10+5*(R-20)
860 CLS:PRINT "There is a chance between
"100-P"% and "
870 PRINT:PRINT 100-PL"% that the "N$(1)
" data"
880 PRINT:PRINT " differs from the "N$(2
)" data"
890 LOCATE 5,23:PRINT "<Space bar for an
other analysis>"
900 A$=INKEY$:IF A$<>" " THEN 900
910 RESTORE:GOTO 40
920 DATA 3.085,2.88,2.75,2.65,2.575,2.51
,2.455,2.41,2.365
930 DATA 2.327,2.052,1.881,1.751,1.645,1
.555,1.476,1.405
940 DATA 1.341,1.282,1.037,.842,.675,.52
4,.385,.253,.126,0
```

7

Space Plan

One of the easiest ways of planning the layout of furniture in a room is to draw on paper an outline of the room and cut out paper shapes to represent the furniture. Then you can slide the shapes around on the plan until you find the most suitable arrangement. This is certainly less tiring than heaving the furniture itself around in the real room! This is also a good way to plan the layout of your garden, or of a workshop or office.

This program makes everything even easier. It works on a similar principle, except that it uses multicoloured displays on the computer screen. You draw an outline of the room or garden on the screen, using the computer's editing keys. Then you can plot circles, rectangles and any other shapes on the screen to represent the furniture or flowerbeds, paths and trees. They can be moved to different parts of the screen. They can be turned around to fit them into place, as required.

The program provides for plans to be saved on tape and loaded back into the computer for immediate display on a later occasion. This could be particularly helpful if you want to prepare several versions of a plan. You can draw a plan and save it. Then you can modify the plan, perhaps moving some of the shapes to different positions, or adding new shapes, and then save the modified version under its own file name.

As well as the applications mentioned above, SPACE PLAN can be used for planning layouts of anything from a stage set to the front page of a magazine. Dressmakers will find that it is easier to move the shapes around the screen than it is to arrange flimsy pattern-pieces on a length of fabric. Rearranging the pieces or adapting the designer's layout to fabric of different width can often save money.

Finally, if you are keen on abstract art, why not use the colourful shapes to create your next masterpiece? Record on tape your own portfolio of computer art.

Using the program

The first message to appear on the screen asks you 'PLANS ON TAPE? (Y/N)'. If you already have a plan saved on tape and wish to load it and perhaps do some more work on it, then press 'Y'. Otherwise press 'N'. If you press 'N' the program continues as described under the heading *Commands*, below.

If you press 'Y', you will then be asked to press PLAY on the recorder, then any key. In a few moments the loading is complete.

Commands

The screen clears to black, with a narrow light blue strip across the bottom. This strip is where messages appear during the running of the program. The black area of the screen is available for your space plan.

The first message to appear is 'BCDMOPRSTXZ?'. This curious combination of letters informs you which keys will produce an effect at this stage. The effect of each key is as follows:

B	Bring back a shape that you have previously deleted.
C	Draw a circle.
D	Delete a shape from the screen.
M	Move a shape.
O	Move all shapes.
P	Draw a polygon.
R	Draw a rectangle.
S	Alter the scale of the plan.
T	Tidy the drawings.
X	Display parameters of shapes.
Z	Save a plan on tape.

The meanings of these commands will become apparent from the descriptions below.

Drawing shapes

The three drawing commands are 'C', 'P' and 'R'. If you key 'C' (no need to press ENTER) a new message appears, asking 'Which one? (0–9)'. You can draw up to ten circles on the screen. Each circle has its own number from 0 to 9. It is as well to note on paper, for future reference, what each circle is intended to represent. In a room plan, a circle might represent a circular table or rug, or perhaps a lamp. In a garden plan a circle could represent a circular garden bed, a pool, or the crown of a tree. Decide what number the circle is to have and press one of the keys, 0 to 9.

The next message to appear asks 'Colour? (1–14)'. In response to this, key a number in this range, depending on which colour you want to use for the circle. The colours are numbered as follows:

1 = Bright yellow
2 = Bright cyan
3 = Bright red
4 = Bright white
5 = Blue
6 = Bright blue
7 = Bright magenta
8 = Cyan
9 = Yellow
10 = Pastel blue
11 = Pink
12 = Bright green
13 = Pastel green
14 = Orange

Select the colour you require and press the key. Then the final message appears. This asks 'What radius?'. In response to this you have to key in a number. Only positive integers (numbers without decimal places) are allowed, but there is no restriction on the size of the number. Normally it should be in the range 5 to 250, but larger and smaller values can be used. The number determines how big the circle is to be. When the program is first run, the width of the black drawing area from left to right is 640 screen units. If you are drawing a plan to scale, you can think of this width as equivalent to 640 centimetres (6.4 metres, the length of a large living-room), or 640 inches (the length of an average garden) or perhaps 64 metres (a large garden). The height of the drawing area is 384 units.

As soon as you have keyed in the radius, press ENTER. The circle will then be drawn on the screen. Drawing a circle is slower than drawing the other shapes but, even so, takes only a few seconds. You will see that the circle has its centre at the centre of the screen. Probably you will not want it in this position, but it is easy to move it elsewhere, as explained later. When the circle is finished, the 'BCDMOPRSTXZ?' message appears again, showing that the computer is waiting for your next command. If you want to draw another circle, repeat the routine described above, choosing a different circle number. You can choose a different colour too, if you wish, and the new circle can have any radius.

The routine for drawing rectangles is very similar to that for drawing circles. You press 'R', select the number of the rectangle, then its colour. After this you are asked to key in the width of the rectangle in screen units. Next you are asked to key in the required height of the rectangle, also in screen units. If you want a square, key in the same height value as you did for width. After you have entered these two values, the rectangle appears at high speed. One of its corners is at the centre of the screen. It can be moved to any position on the screen, and can be rotated to different angles, as explained later.

Polygons are generally used for the more complicated and unusual shapes. A polygon is a shape with any number of straight sides. The program allows the polygons to have up to twelve sides, though each can have a different number of sides. A room with a chimney-piece and a bay window has twelve sides, so a polygon is ideal for drawing the outline of such a room. At the other extreme, a three-sided polygon (or triangle) might represent a corner-cupboard. It is even possible to draw a one-sided polygon, in other words, a straight line! You could use one of these for drawing the door opening into a room, or for drawing a fence in a garden. Taking things to extremes, how about a zero-sided polygon? This appears as a single dot of colour, which might prove useful, though it is rather wasteful of the memory space provided for storing the details of polygons.

Drawing polygons is different from drawing circles and rectangles. You first press 'P' and are then asked to select the polygon number and its colour as explained above. Then you will see a small dot of the specified colour appear near the top left corner of the screen. The message 'Use arrow/copy keys' appears in the message area at the bottom of the screen. Drawing a polygon is done by telling the computer where its corners are to be. As you press the arrow keys (the four at the top right of the keyboard, normally used for editing) a line will be drawn starting from the initial coloured dot (Fig. 7.1). The lines that you draw will *not necessarily* be the sides of the polygon in its final form. Their main purpose is to give you a rough outline. So you need not worry if you are bad at drawing and your line wanders erratically about the screen. The computer will straighten everything later. Also, as you can see, the keys can only draw horizontal or vertical lines, but you may well want some diagonal ones too.

As stated above, the essential point is to mark the positions of the *corners* of the polygon. When the end of the line has reached a point where you want a corner to be, press the COPY key. At once you will see a 'marker' appear. The marker is a short diagonal line, its lower end marking the spot where the corner is to be located on the screen (Fig. 7.2). The marker flashes in contrasting colours, so that you can

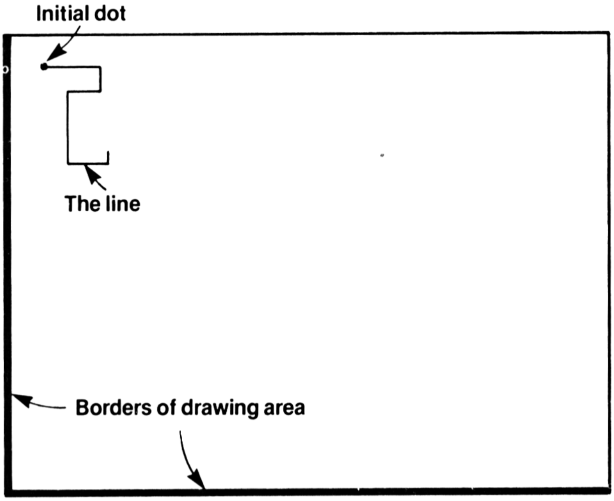


Fig. 7.1. Beginning to draw a polygon.

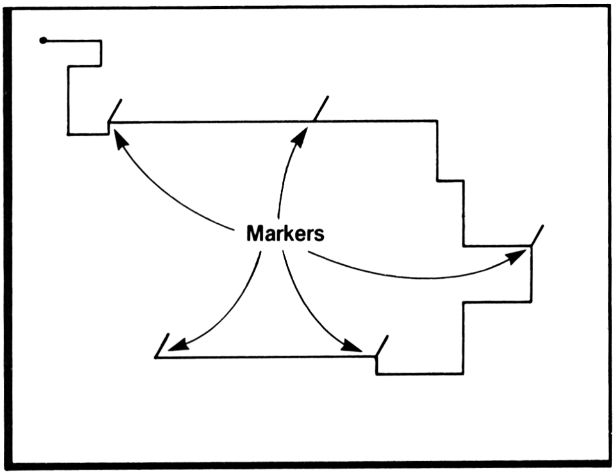


Fig. 7.2. The placing of the markers for a 5-sided polygon.

easily pick it out. Having placed one marker, use the arrow keys to move on to the next corner. Note that you must mark each corner in turn, as if you were walking completely around the room with one hand on the wall, or walking around the boundaries of the garden.

There is no need to complete the final side of the polygon by drawing a line; just mark all its corners and the computer will later join them all up.

When you have marked all the corners, press the space-bar. The screen clears completely, removing all your wandering lines and any other shapes you have already plotted. Then all the shapes reappear: first the circles, followed quickly by the rectangles, and finally the polygon you have just marked (Fig. 7.3). It now consists of straight lines joining the marked points (*without* markers). If you use all twelve markers in drawing a polygon, the clearing and redrawing operations described above take place without you having to press the space-bar first.

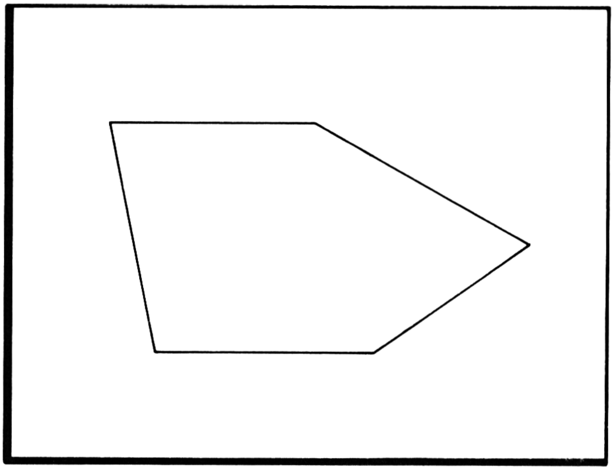


Fig. 7.3. The polygon as drawn by the computer, given the markers placed as in Fig. 7.2.

If you want to draw a room or garden accurately, you may find it hard to gauge the exact places where its corners should be. Figure 7.4 shows how to position the corners accurately. Follow the steps in this figure from A to G.

Although we have dealt with circles, rectangles and polygons in that order, you can of course, select any shape whenever 'BCDMOPRSTXZ' appears.

Deleting and bringing back shapes

If you make a mistake when drawing a circle, rectangle or polygon, it is easy to delete it. If you are in the middle of drawing a polygon, and make a mistake in *marking a corner* (making a mistake in your wandering line does not matter!), press the space-bar. The computer redraws your faulty polygon very neatly, but this is easy to get rid of. When the 'BCDMOPRSTXZ?' message appears, you are able to delete anything on the screen.

To delete a shape, press 'D'. The next message to appear is 'Which shape? (CPR)'. Press 'C', 'P', or 'R', depending on whether you want

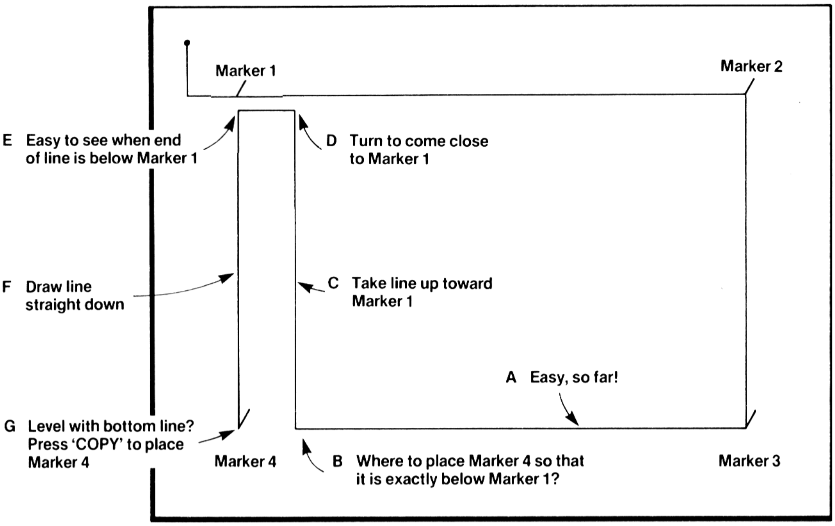

Marker 1
Marker 2
E Easy to see when end of line is below Marker 1
D Turn to come close to Marker 1
F Draw line straight down
C Take line up toward Marker 1
A Easy, so far!
G Level with bottom line? Press 'COPY' to place Marker 4
Marker 4
B Where to place Marker 4 so that it is exactly below Marker 1?
Marker 3

Fig. 7.4. Techniques for placing a marker below or to one side of a marker which is far away. When displayed by the computer, the figure above is a perfect rectangle.

to delete a circle, a rectangle or a polygon. Then you are asked 'Which one? (0–9)'. Key in the number of the shape you want deleted and it instantly disappears. You can then draw another shape to replace the one you have just deleted, giving it the same number as before.

Deletion can be useful in other ways. You may be planning a room and want to know what it will look like without a given item of furniture. Just delete the shape concerned.

The 'bring back' command is the opposite of delete. It brings back to the screen any shape which has been deleted, provided that you have not drawn another shape of the *same number* in the meantime. If circle number 5 was an apple tree and you wanted to see what the garden plan would be like without it, you could delete it. Then you could put it back again by using the 'bring back' command. This command is set up in exactly the same way as the delete command, except that you press 'B' to start with.

Moving a shape

Moving a shape is initiated by pressing 'M' in response to the 'BCDMOPRSTXZ?' message. You are asked to key in the kind of shape ('C', 'P' or 'R') and its number, as with the delete and 'bring back' commands. These commands soon become easy to key in, especially as you do not have to press the RETURN key. For example you key 'D','C','2' in rapid succession to 'Delete Circle 2'. To 'Move Rectangle 6' the keys are 'M', 'R', '6'.

When you have keyed 'M', 'C', '1', for example, to 'Move Circle 1', the message 'Use arrow/copy keys' appears. Pressing the arrow keys moves the shape a short distance across the screen in the direction indicated by the arrow. Press the key very briefly when moving circles. As might be expected, moving a circle for an appreciable distance is a relatively slow operation, but moving the other shapes is very fast. You can move the shapes so that one overlaps the other. You can even move a shape entirely off the screen. This is rather like clearing the furniture out of the room. Provided that you can remember off which side of the screen you moved it, you can bring the shape back on to the screen again later.

Pressing the COPY key causes the shape to rotate. It rotates clockwise. Thirty-six key-presses are needed to turn it completely around. The point about which it rotates is the centre of the drawing area. If you want to rotate a shape that is near the edge of the screen, you may find it easier to move it to the centre of the screen, rotate it the desired amount and then move it back to its position at the edge of the screen. The copy key has no effect on circles.

When you have finished moving or rotating the shape, press the space-bar. This returns you to the 'BCDMOPRSTXZ?' message.

Changing origin

This moves all the shapes the same distance in the same direction. If you have a large plan, not all of which can be seen on the screen at any one time, you can move the whole plan up, down or sideways to bring any part of it into view. By changing scale as well you can 'zoom in' on any part of your plan and work on it in detail, then you can 'zoom in' on another part of the plan to do detailed work there. In this way, a large plan can be examined and worked on in detail.

To change origin, key 'O' (letter 'oh') when the 'BCDMOPRSTXZ?' message appears. You are first asked 'What right?', which is an abbreviated way of asking 'What shift right?'. Key in the number of screen units that you wish the shapes to be moved. To move them to the left, key a negative number. Next you are asked 'What up?'. Key in the number of screen units that you wish the shapes to be moved in an upward direction. To move them down, key a negative number.

The screen clears and there is a pause while the coordinates of the shapes are being re-calculated. If the shapes are still within the part of the plan covered by the screen, you will then see them being drawn again, but in their shifted positions. If you have moved them a long distance, the screen may remain blank. When the shifting process is complete, the 'BCDMOPRSTXZ?' message reappears.

After the existing shapes have been shifted, any new shapes that you draw will appear at the centre of the screen, in the usual way.

Changing scale

The scale of the drawing can be altered at any time. When you press 'S', the message 'Width?' appears. Respond to this by typing in the number of screen units wide you wish the screen to be, from left to right. Initially it is 640, but you can change this to any positive integer value you wish. Type in the number, and press RETURN. There is a short delay while the micro recalculates all the values it has stored. Then the screen clears and the shapes are plotted to the new scale.

If the number you select is greater than the scale previously in use, the objects are displayed smaller than before. As an experiment, having drawn some shapes with the standard 640 units as screen width, change to a screen width of, say, 3000. The shapes reappear on a very small scale clustered in the centre of the screen. You might, for example, have planned the layout of the furniture on the patio; now you can change to another scale (key in a large number) to plan the remainder of the garden. If you change the scale back to its initial value, all the shapes return to their original sizes.

Keying in a small scale number has the effect of magnifying the shapes (i.e. larger sizes). Then you see only those shapes which were previously close to the centre of the screen. You see only parts of other shapes, or perhaps they are completely off-screen. As an example, you can enlarge the shape which represents the dining table, and 'lay the table' with plates, then shrink it again to its initial size. The plates may appear only as dots, but you can easily change the scale again if you want to see them better. If you delete a shape, change scale and then 'bring back' the shape, the shape is plotted to the new scale in its correct position.

Tidying the screen

Moving a shape around the screen, or deleting a shape which overlaps another shape is bound to destroy the continuity of some of the lines. The simple answer to this problem is to use the 'tidy' command. Just press 'T' to make the screen clear and have the shapes perfectly redrawn.

Also use 'T' when you have just loaded a plan from tape. This will cause all previously drawn shapes to appear on the screen.

Displaying parameters

It is often useful to know the exact positions of each of the shapes. By keying 'X' in response to the 'BCDMOPRSTXZ?' message you obtain a display which shows the coordinates of all the shapes. You can use these when drawing an accurate copy of the plan on squared paper.

The screen displays three tables. At the top left is the table of circles. The columns of this table show:

Circle number
X coordinate of its centre
Y coordinate of its centre
R its radius, in screen units

At the top right is the table of rectangles. In this table X1/Y1 to X4/Y4 are the coordinates of the four corners of each rectangle. The bottom half of the screen is occupied by the polygons table. In this table, X1/Y1 to X12/Y12 refer to the coordinates of the twelve corners of each polygon.

At the top centre of the screen, you are shown the current screen width, in screen units.

The tables have room for only three-digit coordinates. If you intend to make use of these tables, it is better to make sure that all shapes are kept within a square which is up to 999 screen units across and that you do not define polygons with negative coordinates (i.e. to the left or below the original screen area). A few extra large or negative values will not matter but, if there are many such values, the tables become difficult to read.

Normally the tables show zeros for the coordinates of shapes that you have not yet defined. If you have shifted the origin (used option 'O'), you will see that the coordinates of undefined shapes have non-zero values. This is a result of the way the shifting routine works. When you define these shapes, their coordinates will be calculated afresh.

Saving on tape

When you key 'Z', the message 'Press REC and PLAY then any key' appears. Wind the tape so that a fresh piece of tape is ready at the recorder head. Then press RECORD and PLAY on the recorder. Finally press any key. The plan is saved under the file name 'PLAN'. You are then offered the chance to make a backup copy. If you decide to do this, press 'Y'. The routine given above is then repeated. If not, press 'N', which returns you to the command message, 'BCDMOPRSTXZ?'

Press T to delete the messages and display the shapes neatly. To begin a new plan, press ESCAPE twice and then rerun the program.

A simple measuring 'tape'

When you draw circles or rectangles, you know the exact measurements of each (in screen units). This is because the computer has asked you to key in their sizes and you have, or should have, made a written note of their sizes. It is not as easy to draw polygons to exact sizes because you are never asked to key in their dimensions. You just draw them out on the screen. You can estimate their dimensions roughly by knowing that the screen normally measures 640 units across and 480 units deep. Another method is to use 'X' to find the coordinates of the corners and then draw the polygon to scale on squared paper.

If you want to draw polygons to an exact size, make yourself a screen scale to help you lay out the polygon accurately. Define a rectangle which is a given number of screen units long. For example, you might define one that is 100 units long. It need be only five units wide. It is better to have two or three, or even more such screen scales. If, for example, you want one side of a polygon to be 400 units long, begin by setting one scale with one of its ends where one end of the side of the polygon is to be. Then use two other scales of the same length, placing them end to end, and 'stepping' them along a distance of 400 units. Leave one of these scales in position to mark where the other end of the side of the polygon is to be. If you want to measure other sides accurately, repeat this process, using more scales if necessary, until the positions of all essential corners have been marked. Press the space-bar when all scales are in position. Then key 'P' to draw a polygon, using these scales to show you where to place the markers. When this is finished, delete all scales and tidy the screen. If you have used several rectangles as scales and have now finished drawing polygons, you can, of course, redefine these rectangles to other sizes, and colours for use as furniture or other objects.

Keying in

The array in which the details of polygons are stored, P(), has three dimensions (see line 30). Whenever you come across this array in the listing, make sure that you type all three numbers or variables that occur between the brackets.

The program uses five text windows. When typing PRINT, INPUT, CLS, LOCATE and WINDOW statements, take care to type the correct figure after the hash (#).

Program design

20–40 Initialising the screen.
50–120 Tape input routine.
130–140 Defining a text window and a graphics window.
150 Coordinates of graphics screen centre.
160–170 Setting colours.
180–310 Waiting for your command.
320–380 Drawing a circle.
390–460 Drawing a rectangle.
470–510 Bringing back a shape.
520–560 Deleting a shape.
570–710 Moving a circle.
720–810 Moving or rotating a rectangle.
820–910 Moving or rotating a polygon.
920–1060 Placing markers when drawing a polygon.
1070–1210 Recalculating values when changing scale.
1220–1350 Redrawing all shapes (used by Tidy, after placing polygon markers and after changing scale).
1360–1490 Shifting origin.
1500–1600 Displaying tables of coordinates.
1610–1770 Saving on tape.
1780–1830 Requests the number of the shape to be drawn, moved, etc.

1840–1880 Requests the code letter for the kind of shape to be drawn, moved, etc.

1890–1940 Requests the size of a shape (radius, etc.) or of screen width when changing scale.

1950–2000 Requests the colour for a shape.

2010–2070 Draws a circle.

2080–2130 Draws a rectangle.

2140–2170 Draws a polygon.

2180–2220 Detects presses on the arrow-keys, when moving a shape and when drawing a polygon.

2230–2310 Recalculates the values of the coordinates when a shape is rotated.

Points of interest

The details of the circles, rectangles and polygons are stored in three arrays, C(), R() and P(), respectively. The colour of each shape is stored as a number from 1 to 14. This number has 16 added to it if the shape is 'active'. That is to say, the shape has been drawn and has not been deleted. When a shape is deleted, 16 is subtracted from the stored value for colour. Thus the details of the shape are still held in the array allowing it to be 'brought back' when required. When this is done, the colour value is increased by 16 again.

The program

```
10 REM ** SPACE PLAN **
20 MODE 0:DEFINT C,N,P,R,X,Y:DEG
30 DIM C(9,3),R(9,10),P(9,12,1)
40 S=1:D$="###"
50 CLS:PRINT"Plans on tape (Y/N)?";
60 A$=UPPER$(INKEY$):IF A$="" THEN 60
70 IF A$="N" THEN 130
80 OPENIN "PLAN"
90 FOR J=0 TO 9:FOR K=0 TO 3:INPUT #9,C(
J,K):NEXT
100 FOR K=0 TO 10:INPUT #9,R(J,K):NEXT
110 FOR K=0 TO 12:FOR L=0 TO 1:INPUT #9,
P(J,K,L):NEXT:NEXT:NEXT
120 CLOSEIN
130 WINDOW #1,1,20,25,25
```

```
140 ORIGIN 0,0,0,639,399,16
150 XC=319:YC=207
160 INK 0,0:INK 5,1:INK 14,15
170 CLG:PAPER #1,10:CLS #1
180 CLS #1:PRINT #1,"BCDMOPRSTXZ?";
190 A$=UPPER$(INKEY$):IF A$="" THEN 190
200 IF A$="B" THEN 470
210 IF A$="C" THEN 320
220 IF A$="D" THEN 520
230 IF A$="M" THEN 570
240 IF A$="O" THEN 1360
250 IF A$="P" THEN 920
260 IF A$="R" THEN 320
270 IF A$="S" THEN 1070
280 IF A$="T" THEN 1220
290 IF A$="X" THEN 1500
300 IF A$="Z" THEN 1610
310 GOTO 180
320 GOSUB 1780
330 GOSUB 1950
340 IF A$="R" THEN 390
350 M$="radius":GOSUB 1890
360 C(N,0)=SZ:C(N,1)=C+16:C(N,2)=XC:C(N,
3)=YC
370 GOSUB 2010
380 GOTO 180
390 M$="width":GOSUB 1890
400 R(N,0)=SZ
410 M$="height":GOSUB 1890
420 R(N,1)=SZ:R(N,2)=C+16:R(N,3)=XC+S*R(
N,0):R(N,4)=YC
430 R(N,5)=R(N,3):R(N,6)=YC-S*R(N,1):R(N
,7)=XC:R(N,8)=R(N,6)
440 R(N,9)=XC:R(N,10)=YC
450 GOSUB 2080
460 GOTO 180
470 GOSUB 1840:GOSUB 1780
480 IF A$="C" AND C(N,1)<16 THEN C(N,1)=
C(N,1)+16:GOSUB 2010
490 IF A$="R" AND R(N,2)<16 THEN R(N,2)=
R(N,2)+16:GOSUB 2080
500 IF A$="P" AND P(N,0,0)<16 THEN P(N,0
,0)=P(N,0,0)+16:GOSUB 2140
510 GOTO 180
520 GOSUB 1840:GOSUB 1780
```

```
530 IF A$="C" AND C(N,1)>15 THEN C(N,1)=
C(N,1)-16:GOSUB 2010
540 IF A$="R" AND R(N,2)>15 THEN R(N,2)=
R(N,2)-16:GOSUB 2080
550 IF A$="P" AND P(N,0,0)>15 THEN P(N,0
,0)=P(N,0,0)-16:GOSUB 2140
560 GOTO 180
570 GOSUB 1840:GOSUB 1780
580 CLS #1:PRINT #1,"Use arrow/copy keys
"
590 IF A$="R" THEN 720
600 IF A$="P" THEN 820
610 X=C(N,2):Y=C(N,3)
620 P$="":WHILE P$<>" "
630 P=0:WHILE (P<240 OR P>243) AND P<>32

640 P$=INKEY$:IF P$="" THEN 640
650 P=ASC(P$)
660 WEND
670 GOSUB 2180
680 PX=C(N,2):PY=C(N,3)
690 IF X<>PX OR Y<>PY THEN C(N,1)=C(N,1)
-16:GOSUB 2010:C(N,2)=X:C(N,3)=Y:C(N,1)=
C(N,1)+16:GOSUB 2010
700 WEND
710 GOTO 180
720 P$="":WHILE P$<>" "
730 P=0:WHILE (P<240 OR P>243) AND P<>32
 AND P<>224
740 P$=INKEY$:IF P$="" THEN 740
750 P=ASC(P$)
760 WEND
770 X=0:Y=0:GOSUB 2180
780 IF X<>0 OR Y<>0 THEN R(N,2)=R(N,2)-1
6:GOSUB 2080:R(N,2)=R(N,2)+16:FOR R=3 TO
 9 STEP 2:R(N,R)=R(N,R)+X:R(N,R+1)=R(N,R
+1)+Y:NEXT:G
OSUB 2080
790 IF P$=CHR$(224) THEN R(N,2)=R(N,2)-1
6:GOSUB 2080:R(N,2)=R(N,2)+16:FOR R=3 TO
 9 STEP 2:X=R(N,R):Y=R(N,R+1):GOSUB 2230
:R(N,R)=X:R(
N,R+1)=Y:NEXT:GOSUB 2080
800 WEND
810 GOTO 180
```

```
820 P$="":WHILE P$<>" "
830 P=0:WHILE (P<240 OR P>243) AND P<>32
 AND P<>224
840 P$=INKEY$:IF P$="" THEN 840
850 P=ASC(P$)
860 WEND
870 X=0:Y=0:GOSUB 2180
880 IF X<>0 OR Y<>0 THEN P(N,0,0)=P(N,0,
0)-16:GOSUB 2140:P(N,0,0)=P(N,0,0)+16:FO
R R=1 TO P(N,0,1):P(N,R,0)=P(N,R,0)+X:P(
N,R,1)=P(N,R
,1)+Y:NEXT:GOSUB 2140
890 IF P$=CHR$(224) THEN P%(N,0,0)=P(N,0
,0)-16:GOSUB 2140:P(N,0,0)=P(N,0,0)+16:F
OR R=1 TO P(N,0,1):X=P(N,R,0):Y=P(N,R,1)
:GOSUB 2230:
P(N,R,0)=X:P(N,R,1)=Y:NEXT:GOSUB 2140
900 WEND
910 GOTO 180
920 GOSUB 1780:GOSUB 1950
930 PN=C:PS=1:P(N,0,0)=C+16
940 CLS #1:PRINT #1,"Use arrow/copy keys
"
950 X=10:Y=390:PLOT X,Y,PN
960 P$="":WHILE PS<13 AND P$<>" "
970 P=0:WHILE (P<240 OR P>243) AND P<>32
 AND P<>224
980 P$=INKEY$:IF P$="" THEN 980
990 P=ASC(P$)
1000 WEND
1010 IF P$=CHR$(224) THEN P(N,PS,0)=X:P(
N,PS,1)=Y:PS=PS+1:DRAW X,Y,PN:DRAWR 25,2
5,15:MOVER -25,-25:GOTO 1040
1020 GOSUB 2180
1030 DRAW X,Y,PN
1040 WEND
1050 P(N,0,1)=PS-1
1060 GOTO 1220
1070 M$="width":GOSUB 1890:SS=S:S=640/SZ
:F=S/SS:CLG
1080 FOR J=0 TO 9
1090 C(J,2)=XC+(C(J,2)-XC)*F
1100 C(J,3)=YC+(C(J,3)-YC)*F
1110 NEXT
1120 FOR J=0 TO 9
```

```
1130 FOR K=3 TO 9 STEP 2
1140 R(J,K)=XC+(R(J,K)-XC)*F
1150 R(J,K+1)=YC+(R(J,K+1)-YC)*F
1160 NEXT:NEXT
1170 FOR J=0 TO 9
1180 FOR K=1 TO 12
1190 P(J,K,0)=XC+(P(J,K,0)-XC)*F
1200 P(J,K,1)=YC+(P(J,K,1)-YC)*F
1210 NEXT:NEXT
1220 CLG
1230 FOR N=0 TO 9
1240 IF C(N,1)<16 THEN 1260
1250 GOSUB 2010
1260 NEXT
1270 FOR N=0 TO 9
1280 IF R(N,2)<16 THEN 1300
1290 GOSUB 2080
1300 NEXT
1310 FOR N=0 TO 9
1320 IF P(N,0,0)<16 THEN 1340
1330 GOSUB 2140
1340 NEXT
1350 GOTO 180
1360 CLS #1:M$="right":GOSUB 1890
1370 XM=SZ
1380 CLS #1:M$="up":GOSUB 1890
1390 YM=SZ
1400 CLG:CLS #1:FOR J=0 TO 9
1410 C(J,2)=C(J,2)+XM:C(J,3)=C(J,3)+YM
1420 NEXT
1430 FOR J=0 TO 9:FOR K=3 TO 9 STEP 2
1440 R(J,K)=R(J,K)+XM:R(J,K+1)=R(J,K+1)+
YM
1450 NEXT:NEXT
1460 FOR J=0 TO 9:FOR K=1 TO 12
1470 P(J,K,0)=P(J,K,0)+XM:P(J,K,1)=P(J,K
,1)+YM
1480 NEXT:NEXT
1490 CLG:GOTO 1220
1500 MODE 2:INK 0,1:INK 1,24:CLS
1510 WINDOW #2, 1,15,1,12:WINDOW #3,25,3
6,4,6:WINDOW #4,46,80,1,12:WINDOW #5,3,7
8,14,25
1520 PRINT #2,"CIRCLES":PRINT #2,"No    X
   Y  Rad";
```

```
1530 FOR J=0 TO 9:LOCATE #2,1,J+3:PRINT
#2,J;:LOCATE #2,5,J+3:PRINT #2, USING D$
;C(J,2);:PRINT #2," ";:PRINT #2, USING D
$;C(J,3);:PR
INT #2," ";:PRINT #2, USING D$;C(J,0);:N
EXT
1540 PRINT #3,"Width =";ROUND(640/S)
1550 PRINT #4,"RECTANGLES":PRINT #4,"No
 X1  Y1  X2  Y2  X3  Y3  X4  Y4"
1560 FOR J=0 TO 9:LOCATE #4,1,J+3:PRINT
#4,J;:FOR K=3 TO 10:PRINT #4, USING D$;R
(J,K);:PRINT #4," ";:NEXT:NEXT
1570 PRINT #5,"POLYGONS":PRINT #5," No
X1 Y1 X2 Y2 X3 Y3 X4 Y4 X5 Y5 X6 Y6 X7 Y
7 X8 Y8 X9 Y9X10Y10X11Y11X12Y12"
1580 FOR J=0 TO 9:LOCATE #5,2,J+3:PRINT
#5,J;:FOR K=1 TO 12:PRINT #5, USING D$;P
(J,K,0);:PRINT #5,CHR$(24);:PRINT #5, US
ING D$;P(J,K
,1);:PRINT #5,CHR$(24);:NEXT:NEXT
1590 A$=INKEY$:IF A$="" THEN 1590
1600 MODE 0:GOTO 130
1610 A$="Y":WHILE A$="Y"
1620 CLS #1
1630 OPENOUT "PLAN"
1640 FOR J=0 TO 9:FOR K=0 TO 3
1650 PRINT #9,C(J,K):NEXT
1660 FOR K=0 TO 10
1670 PRINT #9,R(J,K):NEXT
1680 FOR K=0 TO 12:FOR L=0 TO 1
1690 PRINT #9,P(J,K,L)
1700 NEXT:NEXT:NEXT
1710 CLOSEOUT
1720 A$="":WHILE A$<>"Y" AND A$<>"N"
1730 CLS #1:PRINT #1,"Backup (Y/N)?";
1740 A$=UPPER$(INKEY$):IF A$="" THEN 174
0
1750 WEND
1760 WEND
1770 CLS:GOTO 180
1780 N=-1:WHILE N<0 OR N>9
1790 CLS #1:PRINT #1,"Which one (0-9)?"
1800 N$=INKEY$:IF N$="" THEN 1800
1810 N=VAL(N$)
1820 WEND
```

```
1830 RETURN
1840 A$="":WHILE A$<>"C" AND A$<>"P" AND
 A$<>"R"
1850 CLS #1:PRINT #1,"Which shape (CPR)"
1860 A$=UPPER$(INKEY$):IF A$="" THEN 186
0
1870 WEND
1880 RETURN
1890 SZ=-32768:WHILE SZ<-32767
1900 CLS#1:PRINT #1, "What "M$;:INPUT #1
,S$
1910 IF S$="" OR S$="." THEN 1900
1920 SZ=VAL(S$)
1930 WEND
1940 RETURN
1950 C=0:WHILE C<1 OR C>14
1960 CLS #1:PRINT #1,"Colour (1-14)";:IN
PUT #1,C$
1970 IF C$="" OR C$="." THEN 1960
1980 C=VAL(C$)
1990 WEND
2000 RETURN
2010 R=S*C(N,0)
2020 IF C(N,1)<16 THEN PN=O:ELSE PN=C(N,
1)-16
2030 ORIGIN C(N,2),C(N,3):MOVE R,0
2040 FOR AN=0 TO 360 STEP 6
2050 DRAW R*COS(AN),R*SIN(AN),PN
2060 NEXT:ORIGIN 0,0
2070 RETURN
2080 IF R(N,2)<16 THEN PN=0:ELSE PN=R(N,
2)-16
2090 MOVE R(N,9),R(N,10)
2100 FOR L=1 TO 4
2110 DRAW R(N,1+2*L),R(N,2+2*L),PN
2120 NEXT
2130 RETURN
2140 IF P(N,0,0)<16 THEN PN=0:ELSE PN=P(
N,0,0)-16
2150 MOVE P(N,P(N,0,1),0),P(N,P(N,0,1),1
)
2160 FOR L=1 TO P(N,0,1):DRAW P(N,L,0),P
(N,L,1),PN:NEXT
2170 RETURN
2180 IF P$=CHR$(240) THEN Y=Y+10
```

```
2190 IF P$=CHR$(241) THEN Y=Y-10
2200 IF P$=CHR$(242) THEN X=X-10
2210 IF P$=CHR$(243) THEN X=X+10
2220 RETURN
2230 IF X=XC AND Y=YC THEN RETURN
2240 X=X-XC:Y=Y-YC:Q=SQR(X*X+Y*Y)
2250 IF Y=0 THEN A=180-90*SGN(X):GOTO 22
90
2260 IF X=0 THEN A=90-90*SGN(Y):GOTO 229
0
2270 IF SGN(X)=SGN(Y) THEN A=ATN(X/Y):EL
SE A=180+ATN(X/Y)
2280 IF X<0 THEN A=A+180
2290 A=A+10:IF A>360 THEN A=A-360
2300 X=XC+Q*SIN(A):Y=YC+Q*COS(A)
2310 RETURN
```

There is memory to spare to allow you to increase the number of shapes that can be defined. The only disadvantage of doing this is that the program will take longer to draw the displays. To increase the number of shapes, simply increase the first dimension (the '9') of each array on line 30, to one less than the number of shapes you require. The '9' on lines 90 (only the first), 1080, 1120, 1170, 1230, 1270, 1310, 1400, 1430 (only the first), 1460 and 1650 should be increased too. You will also need to adapt the subroutine on lines 1780–1830 to allow you to key in numbers larger than 9. This means increasing the '9' on line 1780, altering the message, and substituting an INPUT statement on line 1810. The alterations described so far leave the coordinate display routine (option 'X') as it was. It will show only the first 10 shapes of each kind. To show all the additional shapes, this routine would have to be re-written.

If you increase the number of shapes and then receive a 'Memory full' error message when the program is run, try reducing the number of shapes until you no longer receive this message. If you have a printer attached to your micro, routines to print out the display and the table of coordinates would be valuable extensions of the program.

8

Accounts

This is an accounting program suitable for personal use by any member of the family and for small businesses. It can be used for club or committee finances too. It can handle up to eighty different accounts, each identified by name or code number and can deal with up to 3000 separate transactions at one loading. The well known disadvantages of using tape as a storage medium have been minimised by the design of the program. Data is efficiently compacted before recording it, so as to reduce the length of tape used and to cut down the time required for saving and loading.

As well as the ACCOUNTS program, there is a special OPENING program. This is used when you start a new accounting period (including the first time that you use ACCOUNTS). It allows you to begin the period with amounts already held in various accounts. For example, you may already have savings at the bank when you begin your accounting. Or you may have items of equipment or stock that you want to be included in your accounting.

The ACCOUNTS program displays ledger accounts in the familiar format of double-entry bookkeeping. Do not worry if this sounds complicated to you – it is all very simple when it is the computer which has to do the hard work! There are several examples given to make it easy for you to understand how to use the program. The business user will find the trial balance option useful. This takes over what must be one of the worst and most tedious of bookkeeping chores. Once the trial balance has been produced by the computer, it is a relatively easy task for the user to work on the information it contains to produce a trading account, a profit and loss account and a balance sheet.

The program has been designed to do as many useful things as possible, and to handle data efficiently and as fast as possible. The input routines are designed to make it difficult to make mistakes when using the program. It has options which allow you to obtain a

printout of any account or of the trial balance. The result is that this program is one of the longest in the book. This is inevitable if it is to do its job properly.

Using the program

Two tapes are required:

(1) The PROGRAM tape, carrying the OPENING and ACCOUNTS programs.
(2) The AMOUNTS tape, carrying the details of all the accounts (the NAMES and AMOUNTS files).

You may also want to have an additional AMOUNTS tape (or tapes) for backup. A C-10 tape is long enough to hold the maximum number of 3000 transactions.

The instructions below are set out in sections for ease of reference.

(A) Starting your accounts

Before loading the OPENING program, work out on paper the amounts of your opening entries. Figure 8.1 shows opening entries for an account, such as might be used for a club or small business.

The meanings of the words 'credit' and 'debit' need to be explained. When we say that an account is in debit, we mean that there is money in it. When you are paid money (say for a job you have done) you debit your CASH account with an amount. So a *debited* amount is an amount which is *paid in*.

Conversely, an account which is in credit is one out of which you have paid money. If you pay the plumber for mending a burst pipe, you credit your CASH account with that amount. A *credited* amount is an amount that is *paid out*.

When you begin a home budget account you will enter any sums in the bank or in your pocket as accounts in debit. These are amounts which can be made available for you to spend. You could enter all these amounts in one account, called CASH or keep separate accounts for BANK and POCKET. Of course, if you owe the bank money, either because of a loan or overdraft or because you have not paid your credit card account, this item must be entered as a credit on your CASH account. The same applies to any amounts you owe to your friends or for unpaid bills. Should any of your friends owe you money then this amount should be entered as a debit. The balance of these amounts will then be brought down as

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	1500.00	
Cash	700.00	
Balance c/d		2200.00
to Capital a/c		
	2200.00	2200.00

The opening balances of the Equipment account and the Cash accounts are debit. The opening balance of the Capital account is credit.

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	1500.00	
Cash		300.00
Balance c/d		
to Capital a/c		1200.00
	1500.00	1500.00

The opening balance of the Equipment account is debit. The opening balances of the Cash account and the Capital account are credit.

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	348.50	
Cash		1255.00
Balance c/d		
to Capital a/c	906.50	
	1255.00	1255.00

The opening balances of the Equipment account and the Capital account are debit. The opening balance of the Cash account is credit.

Fig. 8.1. Basic examples of opening entries.

printout of any account or of the trial balance. The result is that this program is one of the longest in the book. This is inevitable if it is to do its job properly.

Using the program

Two tapes are required:

(1) The PROGRAM tape, carrying the OPENING and ACCOUNTS programs.
(2) The AMOUNTS tape, carrying the details of all the accounts (the NAMES and AMOUNTS files).

You may also want to have an additional AMOUNTS tape (or tapes) for backup. A C-10 tape is long enough to hold the maximum number of 3000 transactions.

The instructions below are set out in sections for ease of reference.

(A) Starting your accounts

Before loading the OPENING program, work out on paper the amounts of your opening entries. Figure 8.1 shows opening entries for an account, such as might be used for a club or small business.

The meanings of the words 'credit' and 'debit' need to be explained. When we say that an account is in debit, we mean that there is money in it. When you are paid money (say for a job you have done) you debit your CASH account with an amount. So a *debited* amount is an amount which is *paid in.*

Conversely, an account which is in credit is one out of which you have paid money. If you pay the plumber for mending a burst pipe, you credit your CASH account with that amount. A *credited* amount is an amount that is *paid out.*

When you begin a home budget account you will enter any sums in the bank or in your pocket as accounts in debit. These are amounts which can be made available for you to spend. You could enter all these amounts in one account, called CASH or keep separate accounts for BANK and POCKET. Of course, if you owe the bank money, either because of a loan or overdraft or because you have not paid your credit card account, this item must be entered as a credit on your CASH account. The same applies to any amounts you owe to your friends or for unpaid bills. Should any of your friends owe you money then this amount should be entered as a debit. The balance of these amounts will then be brought down as

8

Accounts

This is an accounting program suitable for personal use by any member of the family and for small businesses. It can be used for club or committee finances too. It can handle up to eighty different accounts, each identified by name or code number and can deal with up to 3000 separate transactions at one loading. The well known disadvantages of using tape as a storage medium have been minimised by the design of the program. Data is efficiently compacted before recording it, so as to reduce the length of tape used and to cut down the time required for saving and loading.

As well as the ACCOUNTS program, there is a special OPENING program. This is used when you start a new accounting period (including the first time that you use ACCOUNTS). It allows you to begin the period with amounts already held in various accounts. For example, you may already have savings at the bank when you begin your accounting. Or you may have items of equipment or stock that you want to be included in your accounting.

The ACCOUNTS program displays ledger accounts in the familiar format of double-entry bookkeeping. Do not worry if this sounds complicated to you – it is all very simple when it is the computer which has to do the hard work! There are several examples given to make it easy for you to understand how to use the program. The business user will find the trial balance option useful. This takes over what must be one of the worst and most tedious of bookkeeping chores. Once the trial balance has been produced by the computer, it is a relatively easy task for the user to work on the information it contains to produce a trading account, a profit and loss account and a balance sheet.

The program has been designed to do as many useful things as possible, and to handle data efficiently and as fast as possible. The input routines are designed to make it difficult to make mistakes when using the program. It has options which allow you to obtain a

printout of any account or of the trial balance. The result is that this program is one of the longest in the book. This is inevitable if it is to do its job properly.

Using the program

Two tapes are required:

(1) The PROGRAM tape, carrying the OPENING and ACCOUNTS programs.
(2) The AMOUNTS tape, carrying the details of all the accounts (the NAMES and AMOUNTS files).

You may also want to have an additional AMOUNTS tape (or tapes) for backup. A C-10 tape is long enough to hold the maximum number of 3000 transactions.

The instructions below are set out in sections for ease of reference.

(A) Starting your accounts

Before loading the OPENING program, work out on paper the amounts of your opening entries. Figure 8.1 shows opening entries for an account, such as might be used for a club or small business.

The meanings of the words ‘credit’ and ‘debit’ need to be explained. When we say that an account is in debit, we mean that there is money in it. When you are paid money (say for a job you have done) you debit your CASH account with an amount. So a *debited* amount is an amount which is *paid in*.

Conversely, an account which is in credit is one out of which you have paid money. If you pay the plumber for mending a burst pipe, you credit your CASH account with that amount. A *credited* amount is an amount that is *paid out*.

When you begin a home budget account you will enter any sums in the bank or in your pocket as accounts in debit. These are amounts which can be made available for you to spend. You could enter all these amounts in one account, called CASH or keep separate accounts for BANK and POCKET. Of course, if you owe the bank money, either because of a loan or overdraft or because you have not paid your credit card account, this item must be entered as a credit on your CASH account. The same applies to any amounts you owe to your friends or for unpaid bills. Should any of your friends owe you money then this amount should be entered as a debit. The balance of these amounts will then be brought down as

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	1500.00	
Cash	700.00	
Balance c/d to Capital a/c		2200.00
	2200.00	2200.00

The opening balances of the Equipment account and the Cash accounts are debit. The opening balance of the Capital account is credit.

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	1500.00	
Cash		300.00
Balance c/d to Capital a/c		1200.00
	1500.00	1500.00

The opening balance of the Equipment account is debit. The opening balances of the Cash account and the Capital account are credit.

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	348.50	
Cash		1255.00
Balance c/d to Capital a/c	906.50	
	1255.00	1255.00

The opening balances of the Equipment account and the Capital account are debit. The opening balance of the Cash account is credit.

Fig. 8.1. Basic examples of opening entries.

the opening balance on your home budget account.

If you simply want to start your home account from scratch, you must use the program to enter a single account. You might call it 'H'KPING' (for 'housekeeping') or 'CASH'. Start it with '0.00' as the amount. You can type 'C' or 'D' in answer to the 'Debit or Credit' question.

Opening an account as a business is similar, though there are special names given to certain types of account. If you are starting a new business or other enterprise, the amounts to be accounted for will usually come under three headings (see Fig. 8.1):

In the CASH account: if you have money in the bank this is a debit on the opening balance. If you start with a bank overdraft this is a credit on the opening balance. Remember, a debit is a sum which is owed to you. When you put money in the bank, the bank then owes you that money.

In the EQUIPMENT account: the value of equipment, stock or other items which you already have when the business is started. This is a debit item, in the sense that you might later sell part of the equipment, and the money for this sale would be owed to the business.

In CAPITAL account: this must equal the total of the opening balances of the CASH and EQUIPMENT accounts. It represents what the business owes to the proprietor or to those who have lent money to start the business. Figure 8.1 will help you decide if this is a credit or debit balance.

(B) The OPENING program

Load and run the OPENING program. Note that the programs are able to accept account names which are typed in capital letters (up to 7 letters) (e.g. CASH), or in a mixture of capital and lower-case letters (e.g. Cash).

The OPENING program asks you for the date (of opening the accounts), the account names, the opening amounts and whether they are debits or credits. Section G explains how to enter dates; section H explains how to enter amounts. As you finish entering the details of each account, you are asked 'ALL OK? (Y/N)'. If you are satisfied that you have typed in everything correctly, key 'Y'. Otherwise, key 'N' and you are asked to enter all the details of that account again.

Typing 'Y' makes the micro open an account with the details you have typed in. After each account has been opened, you are asked 'More?'. Key 'Y' to open another account; key 'N' when you have opened all the accounts you need to start with. There is no need to enter accounts which begin with zero balances, since this is done in the main ACCOUNTS program. The OPENING program allows you to start with between one and nine accounts.

When you have typed in the last account details and have pressed 'N' to indicate that no more accounts are required or when you have typed in details of nine accounts, the message 'Press REC and PLAY then any key' appears on the screen. Insert the AMOUNTS tape in the recorder and rewind or fast forward the tape so that it is ready for recording. Press REC and PLAY on the recorder, then any key on the micro. The message 'Saving NAMES block 1' appears and the recorder runs for about ten seconds. A second 'Press REC and PLAY then any key' appears. Press any key on the keyboard. The next message is 'Saving AMOUNTS block 1'. The number at the end of this message is incremented regularly. After it has reached 11, the recorder stops. This operation takes about five minutes. Two files have now been prepared on tape, ready to hold all information that you may wish to store there in future. The OPENING program has now finished.

(C) Starting the ACCOUNTS program

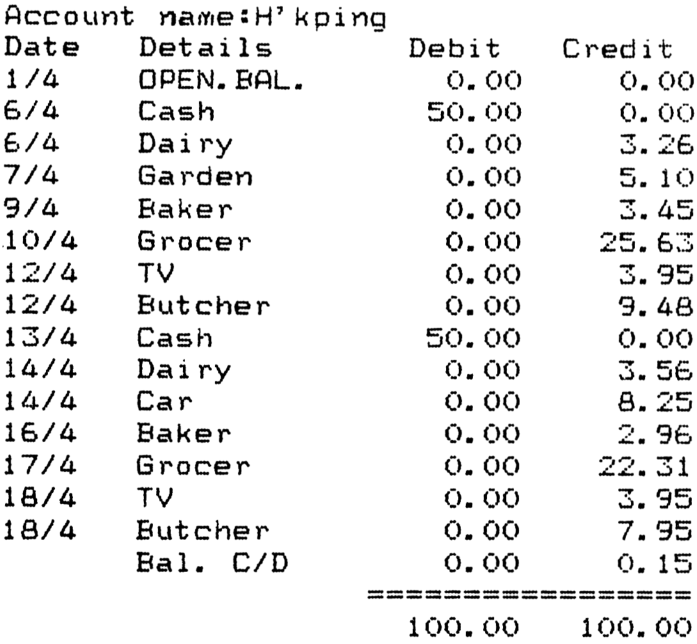
Now LOAD and RUN the ACCOUNTS program. You are first asked to press the 'Space Bar when AMOUNTS tape ready'. If you have not already done so, remove the PROGRAM tape from the recorder and replace it with the AMOUNTS tape. Then press the space-bar. The message 'Press PLAY then any key' appears. After you have obeyed this instruction you see the message 'Loading NAMES block 1'. About ten seconds later the 'Press PLAY then any key' message is repeated. Press any key. The message 'Loading AMOUNTS block 1' is displayed next. The number at the end of the message is incremented until it reaches 11. This takes about five minutes, at the end of which the recorder stops and the screen displays the menu.

(D) Using the menu

The menu invites you to choose between seven options but, if you have entered only one account when you used OPENING, you can select only Option 3 at this stage. If this is the case, you are reminded of it by the fact that only Option 3 is displayed in white at this stage. Obviously the program needs at least two accounts on its records before it can start to transfer sums from one account to another.

The menu is the central point in the operation of the program. Whenever you select an option (by pressing the key which has the same number as the option) you are sent to the part of the program which performs that task. Afterwards, (unless you select '7 Finish'), you are returned to the menu to decide what you want the computer to do next. The options work as follows:

(1) *Display account:* A list of the account names or codes is displayed and you are asked to choose any one by typing its number, as listed. You are also asked to type in the number of the month from which you wish the display to begin. Transactions are displayed from the first day of that month. You are then asked to type in the number of the month up to which the account is to be displayed. It will be displayed up to the last day of that month. If you want to see everything on the account, simply type '1' as the 'From' month and '12' as the 'To' month. The account is then displayed, showing the opening balance (if any), every transaction and its date, and amounts debited or credited and the balance carried down at the end of the period. If there are many transactions in the account, press ESC to stop the display when the screen is full. Press the space-bar to view the next screenful of transactions.

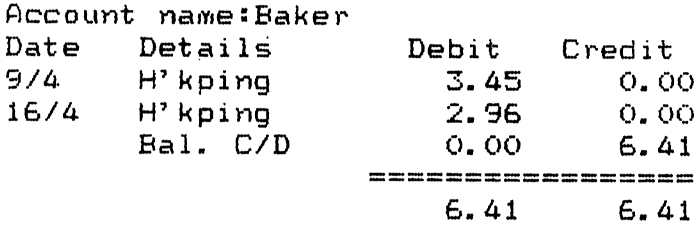


```
Account name:H'kping
Date    Details          Debit    Credit
1/4     OPEN.BAL.         0.00      0.00
6/4     Cash             50.00      0.00
6/4     Dairy             0.00      3.26
7/4     Garden            0.00      5.10
9/4     Baker             0.00      3.45
10/4    Grocer            0.00     25.63
12/4    TV                0.00      3.95
12/4    Butcher           0.00      9.48
13/4    Cash             50.00      0.00
14/4    Dairy             0.00      3.56
14/4    Car               0.00      8.25
16/4    Baker             0.00      2.96
17/4    Grocer            0.00     22.31
18/4    TV                0.00      3.95
18/4    Butcher           0.00      7.95
        Bal. C/D          0.00      0.15
                         ================
                        100.00    100.00
```

Fig. 8.2. Mrs Smith's housekeeping account.

Figure 8.2 shows Mrs Smith's housekeeping account for two weeks.

(2) *Transaction:* Home users may be unfamiliar with terms such as 'credit balance', and 'double-entry' accounting. The essence of this accounting system is that money always comes *out of* somewhere and goes *into* somewhere else. We need an account from which money is taken (this account is credited) and an account into which the money is paid (the account which is debited). For example, if Mrs Smith pays the baker for a loaf of bread, she needs an account called 'Baker' and an account called 'Housekeeping'. The 'Baker' account is debited with the cost of the loaf, while the 'Housekeeping' account is credited with the same amount. These two operations are performed together by the computer when you use Option 2. If you now display both accounts, her 'housekeeping' account will look like Fig. 8.2 and her 'Baker' account will look like Fig. 8.3. You have to open an account for each different kind of item you spend money on. This makes it very easy to check just how much you spend with people such as the butcher, the baker and the electricity board. The program can hold up to eighty different accounts, so allowing you to analyse your expenditure in detail. However, there are some items which could be grouped together under one main heading. For example you would not want separate accounts for shoes, socks, jeans, and hats, but would charge all of these to a single account called 'Clothes'. Figure 8.4 shows a typical cash account for a small business.



```
Account name:Baker
Date     Details          Debit    Credit
9/4      H'kping           3.45      0.00
16/4     H'kping           2.96      0.00
         Bal. C/D          0.00      6.41
                    ==================
                           6.41      6.41
```

Fig. 8.3. Mrs Smith's account for the baker.

When you select Option 2, a numbered list of account names is displayed. You are asked to key in the number of the account which is to be credited (i.e. the account out of which the sum of money is to be paid) and the number of the account to be debited (i.e. the account into which the money is to be paid). You are also asked the

```
Account name:Cash
Date    Details          Debit    Credit
1/1     OPEN.BAL.       700.00      0.00
5/1     XYZ Ltd         250.00      0.00
7/1     Browns          163.00      0.00
8/1     Vehicle           0.00     35.70
8/1     Fuel              0.00    116.23
10/1    ABC Co           17.32      0.00
12/1    XYZ Ltd         350.00      0.00
15/1    VAT               0.00    175.82
20/1    Bank Ch           0.00     15.73
21/1    Phone             0.00     78.77
23/1    Browns          532.95      0.00
23/1    Vehicle           0.00     13.58
30/1    Wages             0.00    233.00
30/1    Drawngs           0.00    500.00
        Bal. C/D          0.00    844.44
                      ================
                       2013.27   2013.27
```

Fig. 8.4. Small business cash account using the first opening entries of Fig. 8.1.

date of the transaction (see section G). If your accounts begin from January of each year, enter the day of the month ('01' to '29', '30' or '31') followed immediately (no space or punctuation) by the month number ('01' to '12'). If your accounts begin in some other month and you want to keep accounts for a whole year, enter '01' for the first month in your accounts year, '02' for the second month, and so on. Transactions should be entered in order of date as far as possible. The essential point is that all transactions taking place in a given month must be entered before you enter any transactions of the following month.

If, on looking at the list of accounts, you find that one of the accounts is not on the list, key '0' as the account number. This takes you direct to the 'Open new account' option, so that you can open the 'missing' account, as described in the next section.

(3) *Open new account:* The first question is 'What is the name of the new account?'. Key this in, using all capitals or both capitals and lower-case letters as preferred. Note that account names can be only seven characters long. Any characters in excess of this are chopped off by the computer. Account names, though referred to as such in the program and in this description, need not be names. You can use codes made up of letters and numerals, or even solely of numerals.

They can include punctuation marks, except for commas. Press ENTER after you have typed in the account name. The computer searches its memory to discover if an account of that name has already been opened. If so, it informs you of this fact and then returns you to the menu. If the account is not open already, it displays the name of the account again, so that you can confirm that you have typed it in correctly. If the name you typed had more than seven characters, only the first seven are displayed. You are then asked to confirm that the name is correct. If so, you are returned to the menu. If not, you are asked to enter it again.

(4) *Print account:* This does the same as Option 1, except that instead of displaying the account on the screen, the account is printed by a printer attached to the micro. Figures 8.2 to 8.4 were prepared by using this option.

(5) *Display trial balance:* This covers the period from the beginning of the accounting period until the end of any chosen month. Figure 8.5 shows a trial balance prepared from the accounts of Fig. 8.4. Accounts which have zero balance at the end of the period, or on which there have been no transactions during the period do not appear on the trial balance.

```
Trial balance to end of month 1
A/c     Name          Debit     Credit
1        Equipt      1500.00       0.00
2        Cash         844.44       0.00
3        Capital        0.00    2200.00
4        XYZ Ltd        0.00     600.00
5        Browns         0.00     695.95
6        ABC Co         0.00      17.32
7        Vehicle       49.28       0.00
8        VAT          175.82       0.00
9        Bank Ch       15.73       0.00
10       Phone         78.77       0.00
11       Wages        233.00       0.00
13       Fuel         116.23       0.00
15       Drawngs      500.00       0.00
                   ==================
                     3513.27    3513.27
```

Fig. 8.5. Trial balance of the business accounts illustrated in Fig. 8.4. Account numbers 12 and 14 had not shown any transactions during the period.

(6) *Print trial balance:* This is the same as Option 5, except that you obtain a printout instead of a screen display.

(7) *Finish:* It is essential to use this option at the end of every session using the program. If you simply switch off, all data entered during the current session is lost.

The first message to appear is 'Finishing', followed shortly by 'Press REC and PLAY then any key'. Rewind the AMOUNTS tape to the beginning, so that it is ready to record the new data over the old tracks. Then press REC and PLAY on the recorder, followed by any key on the computer. The message 'Loading NAMES block 1' appears. After about ten seconds, 'Press REC and PLAY then any key' is displayed. REC and PLAY are already pressed, so just press any key on the keyboard. Now the message 'Saving AMOUNTS block 1' appears. The number is incremented steadily until it reaches 11. At this point the recorder stops. This process takes five minutes. The length of time is fixed, and does not depend on how many accounts there are. It will take no longer even when you have as many as 3000 transactions in memory.

When the recorder stops you are offered the option of taking a backup copy of the transactions. If you decide to do this, which is advisable, put another tape labelled 'AMOUNTS' in the recorder. Alternatively, you can record the backup on the same tape, immediately after the first recording. If you key 'Y', the procedure described above is repeated. If you press 'N', the program ends. It is now safe to switch off the computer, rerun ACCOUNTS from the beginning, or load and run another program. Before loading another program, press CTRL, SHIFT and ESC, to make the full amount of memory available for the new program.

(E) Using ACCOUNTS again

The procedure is exactly the same as in section (C), above. Then use the program exactly as described in section (D) above.

(F) Using OPENING again

This program is used whenever you want to start a new set of accounts. You will need to use it after your accounts have been open for a full year, or when the number of transactions reaches 3000.

(G) Typing dates

The format used for dates requires four figures:

DDMM

where DD are the figures for the day of the month and MM are the figures for the number of the month. Thus 1st April is represented by '0104', and 21 November by '2111'. If your accounting year does not begin on 1st January, you may number the months differently, as described in section (D) (2).

Although the program does not prevent you from entering imaginary dates such as 31st February, it does check that the days are not more than 31 and the month is not more than 12. If the message is repeated when a date is typed in, you have typed in an illegal date.

(H) Typing amounts

Sums of money are entered as:

up to five digits, for the pounds
a full stop (point)
two digits, for the pence

There is no need to press ENTER after entering the second pence digit; the program continues immediately. If you have entered 5 'pounds' digits, the program will not accept anything other than a full stop, then the two 'pence' digits. Examples of acceptable amounts are:

£30305.45
£555.06
£67.00

The largest sum that can be entered is £65535.99. If you try to enter a sum larger than this amount, the computer will wait until it has been overtyped to correct the error.

(I) Getting out of trouble

Although care has been taken to make it difficult or even impossible for the computer to accept incorrect entries, there are still occasions on which you may press the wrong key by mistake. The hints below will help you recover without upsetting the program:

(1) You keyed Options 1 or 2 and then found that the account you thought had been opened had not already been opened. Key '0' when asked for the account number required. This takes you directly to Option 3, to open the 'missing' account.

(2) You have keyed Option 3 in mistake for one of the other options, and have no new account to be opened. Type the name of an account which you know is already open. The computer will then tell you that this account is already open and, after a short pause, send you back to the menu.

(3) You have made a mistake in entering details of a transaction and it is already on the account concerned. There is no way of deleting transactions from the accounts. Instead, follow the usual bookkeeping practice of making a *contra entry*. For example, if you entered a payment of £25.45 from the cash account to the fuel account, when it should have been £25.54, proceed as follows. Make a payment of £25.45 from the fuel account into the cash account (this is the contra entry). Then make a payment of the correct amount (£25.54) from the cash account into the fuel account.

(4) You accidentally press the ESCAPE key twice and halt the ACCOUNTS program. Key in 'GOTO 170' and press ENTER. This will take you back to the menu without loss of data.

Keying in

In OPENING, line 100, there are seven spaces between the quotes. In ACCOUNTS, take particular care with the colons and semicolons, for errors with these can ruin the display. For the same reason, be sure to key in the correct number of 'PRINT's in the PRINT statements. Mistyping of some of the numbers could cause chaos in both programs when they are run. Numbers to take especial care with are those in lines 30 and 430 of OPENING and in lines 30, 150,1300,1550, 1670 and 1890 of ACCOUNTS. Letter 'oh' is not often used in variable names, but we use it in OPENING, in line 80 and in ACCOUNTS, in lines 1190–1220, in the variables MO$ and MO.

Program designs

OPENING

20–40 Initialising, including protecting memory to receive the data (line 30).

50–80 Inputting and checking the date.

90–100 Inputting the account name and shortening it if necessary.

110–210 Inputting and checking the amount.

220–240 Credit or debit?
250–270 Confirming entries.
280–340 Storing details in memory.
350–380 More accounts?
390–440 Saving NAMES and AMOUNTS files.

ACCOUNTS
20–60 Initialising.
70–90 Waiting to load from tape.
100–150 Loading from tape.
160 Setting colours.
170–280 Displaying the menu.
290–330 Selection of options.
340–380 Displaying the list of account names.
390 Branch to input transaction details.
400–500 Details for account to be displayed.
510–560 Calculating the balance of an account for any period prior to the period for which it is to be displayed (= balance brought down).
570–670 Displaying or printing headings of account and balance brought down.
680–780 Searching memory for transactions on that account, and displaying or printing them, accumulating credit and debit totals.
790–910 Calculating balance carried down and total balances and displaying or printing them.
920–940 Return to menu.
950–980 Inputting name of new account.
990–1020 Searching to see if an account in this name already exists.
1030–1100 Confirming account name.
1110–1230 Inputting transaction details; account numbers and date.
1240–1350 Inputting amount.
1360–1450 Confirming that details of transactions are correct.
1460–1480 Storing transaction details in memory.
1490–1500 Return to menu, with warning if memory is nearly full.
1510–1540 Inputting period for trial balance.
1550–1620 Displaying or printing headings for trial balance.
1630–1830 Calculating and displaying or printing trial balances.
1840–1890 Saving names and details to tape.
1900–1940 Optional backup.
1950–1960 Delay to allow user time to read warning messages.

Points of interest

The names of accounts are held in the array N$( ). They are saved in a file 'NAMES' preceded by N, the number of accounts and NR, the number of transactions.

The details of transactions are stored directly in memory in blocks of seven bytes. Set aside for this purpose are 21000 bytes of memory, allowing 3000 transactions to be stored in a very compact form. This block of memory extends from address 20000 to 41000. The contents of each block of seven bytes are:

Byte 0	The number of the account credited.
Byte 1	The number of the account debited.
Byte 2	The day.
Byte 3	The month.
Byte 4	The number of pounds (high byte).
Byte 5	The number of pounds (low byte).
Byte 6	The number of pence.

The actual number of pounds is found by multiplying the contents of byte 4 by 256 and then adding the contents of byte 5.

The program (OPENING)

```
10 REM ** OPENING BALANCES **
20 MODE 1
30 MEMORY 19999
40 DIM N$(9):N=1
50 CLS:LOCATE 1,5:INPUT "Date (DDMM)? ",
D$
60 IF LEN(D$)<>4 THEN 50
70 DA$=LEFT$(D$,2):IF VAL(DA$)<1 OR VAL(
DA$)>31 THEN 50
80 MO$=RIGHT$(D$,2):IF VAL (MO$)<1 OR VA
L(MO$)>12 THEN 50
90 LOCATE 1,7:INPUT "Account name (7 cha
rs)? ",N$(N)
100 N$(N)=LEFT$(N$(N)+"       ",7)
110 NK=0:AM$="":LOCATE 1,9:PRINT "Amount
 (pounds.pence)? ";
120 A$=INKEY$:IF A$="" THEN 120
130 A=ASC(A$):IF A<46 OR A>57 OR A=47 TH
EN 120
140 IF NK=5 AND A$<>"." THEN 120
150 PRINT A$;:IF A$="." THEN 170
160 AM$=AM$+A$:NK=NK+1:IF NK<6 THEN 120
```

```
170 IF VAL(AM$)>65535 THEN 110
180 NK=0:PE$=""
190 A$=INKEY$:IF A$="" THEN 190
200 A=ASC(A$):IF A<48 OR A>57 THEN 190
210 PRINT A$;:PE$=PE$+A$:NK=NK+1:IF NK<2
 THEN 190
220 PRINT:PRINT:PRINT "Opening balance d
ebit or credit (D/C) ";
230 B$=UPPER$(INKEY$):IF B$<>"C" AND B$<
>"D" THEN 230
240 PRINT B$
250 PRINT:PRINT "All OK?  (Y/N)";
260 A$=UPPER$(INKEY$):IF A$<>"Y" AND A$<
>"N" THEN 260
270 IF A$="N" THEN N=1:GOTO 50
280 U=20000+7*(N-1)
290 IF B$="C" THEN POKE U,N:POKE U+1,0
300 IF B$="D" THEN POKE U,0:POKE U+1,N
310 POKE U+2,VAL(LEFT$(D$,2)):POKE U+3,V
AL(RIGHT$(D$,2))
320 POKE U+4,INT(VAL(AM$)/256)
330 POKE U+5,VAL(AM$)-PEEK(U+4)*256
340 POKE U+6,VAL(PE$)
350 IF N=9 THEN 390
360 PRINT:PRINT:PRINT "More?  (Y/N)"
370 A$=UPPER$(INKEY$):IF A$<>"Y" AND A$<
>"N" THEN 370
380 IF A$="Y" THEN N=N+1:CLS:GOTO 90
390 PRINT:OPENOUT "NAMES"
400 PRINT #9,N:PRINT #9,N
410 FOR J=1 TO N:PRINT #9,N$(J):NEXT
420 CLOSEOUT
430 SAVE "AMOUNTS",B,20000,21000
440 END
```

The program (ACCOUNTS)

```
10 REM ** ACCOUNTS **
20 MODE 1
30 MEMORY 19999
40 DIM N$(80)
50 N=0:JJ=2:C$="#####.##"
60 FOR J=1 TO 80:N$(J)=STRING$(8," "):NE
XT
70 LOCATE 1,5:PRINT "Space bar when AMOU
NTS tape ready"
```

```
80 A$=INKEY$
90 IF A$<>" " THEN 80
100 OPENIN "NAMES"
110 INPUT #9,N
120 INPUT #9,NR
130 FOR J=1 TO N:INPUT #9,N$(J):NEXT
140 CLOSEIN
150 LOAD "AMOUNTS",20000
160 INK 1,20:INK 2,8:INK 3,26
170 CLS
180 PEN 1:LOCATE 12,3:PRINT "** ACCOUNTS
 **"
190 PRINT TAB(15) "--------"
200 MC%=3:IF N<2 THEN MC%=1
210 PEN MC%:PRINT:PRINT:PRINT TAB(6)"1)
 Display account"
220 PEN MC%:PRINT:PRINT TAB(6)"2)  Trans
action"
230 PEN 3:PRINT:PRINT TAB(6)"3)  Open ne
w account"
240 PEN MC%:PRINT:PRINT TAB(6)"4)  Print
 account"
250 PEN MC%:PRINT:PRINT TAB(6)"5)  Displ
ay trial balance"
260 PEN MC%:PRINT:PRINT TAB(6)"6)  Print
 trial balance"
270 PEN MC%:PRINT:PRINT TAB(6)"7)  Finis
h"
280 PEN 2:PRINT:PRINT:PRINT TAB(3) "<Sel
ect by pressing numbered key>";
290 A$=INKEY$:IF A$="" THEN 290
300 A=VAL(A$):IF A<0 OR A>8 THEN 290
310 IF NR=1500 AND A=2 THEN CLS:BORDER 6
,9:LOCATE 9,11:PRINT "There is no more r
oom":GOSUB 1960:BORDER 1:GOTO 170
320 IF N<2 AND A<>3 THEN 290
330 ON A GOTO 340,340,950,340,1510,1510,
1840
340 CLS
350 JJ=2:FOR J=1 TO 20
360 IF J>9 THEN JJ=1
370 PEN 3:LOCATE JJ,J:PRINT MID$(STR$(J)
,2);N$(J);:LOCATE 10,J:PRINT STR$(J+20);
N$(J+20);:LOCATE 20,J:PRINT STR$(J+40);N
$(J+40);:LOC
ATE 30,J:PRINT STR$(J+60);N$(J+60)
```

```
380 NEXT
390 IF A=2 THEN 1110
400 NA=-1:WHILE NA<0 OR NA>N
410 LOCATE 10,21:PRINT "Account no.";:IN
PUT;NA$
420 NA=VAL(NA$)
430 WEND
440 IF NA=0 THEN 950
450 SM=-1:WHILE SM<0 OR SM>13
460 LOCATE 1,22:INPUT "From which month
(1-12)";SM$:SM=VAL(SM$)
470 WEND
480 FM=-1:WHILE FM<SM OR FM>13
490 LOCATE 1,23:INPUT "To which month (1
-12)";FM$:FM=VAL(FM$)
500 WEND
510 J=0:B=0
520 U=20000+7*J
530 IF PEEK(U+3)>=SM THEN 570
540 IF PEEK(U)=NA THEN B=B-PEEK(U+4)*256
-PEEK(U+5)-PEEK(U+6)/100
550 IF PEEK(U+1)=NA THEN B=B+PEEK(U+4)*2
56+PEEK(U+5)+PEEK(U+6)/100
560 J=J+1:IF J<NR THEN 520
570 CLS
580 PEN 1:IF A=4 THEN PRINT:PRINT:PRINT
"Printing account":P=8
590 PRINT #P, "Account name:"N$(NA)
600 CD=0:DD=0
610 PRINT:PRINT #P,"Date   Details
 Debit   Credit":PRINT:Y=5
620 IF B=0 THEN C=0:D=0:GOTO 680
630 IF B<0 THEN D=0:C=-B
640 IF B>0 THEN D=B:C=0
650 TR$="Balance B/D":CD=C:DD=D
660 PEN 3:IF P=0 THEN PRINT "1/"SM$;:LOC
ATE 7,5:PRINT TR$:LOCATE 20,5:PRINT USIN
G C$;DD;:PRINT " ";:PRINT USING C$;CD:Y=
Y+1
670 IF P=8 THEN PRINT#8,"1/"SM$;TAB(7)TR
$;PRINT TAB(20) USING C$;DD;PRINT " ";PR
INT USING C$;CD
680 U=20000+7*J
690 IF PEEK(U+3)>=FM+1 THEN 790
700 IF PEEK(U)<>NA AND PEEK(U+1)<>NA THE
N 780
```

```
710 M=PEEK(U+4)*256+PEEK(U+5)+PEEK(U+6)/
100
720 IF PEEK(U)=NA THEN C=C+M:CD=M:DD=0:T
R$=N$(PEEK(U+1))
730 IF PEEK(U+1)=NA THEN CD=0:D=D+M:DD=M
:TR$=N$(PEEK(U))
740 IF PEEK(U)=0 OR PEEK(U+1)=0 THEN TR$
="Open.Bal."
750 DA$=MID$(STR$(PEEK(U+2)),2)+"/"+MID$
(STR$(PEEK(U+3)),2)
760 PEN 3:IF P=0 THEN PRINT DA$;:LOCATE
7,Y:PRINT TR$:LOCATE 20,Y:PRINT USING C$
;DD;:PRINT " ";:PRINT USING C$;CD:Y=Y+1
770 IF P=8 THEN PRINT #8, DA$;TAB(7)TR$;
:PRINT #8,TAB(20) USING C$;DD;:PRINT #8,
" ";:PRINT #8,USING C$;CD
780 J=J+1:IF J<NR THEN 680
790 B=C-D
800 IF B=0 THEN 850
810 IF B<0 THEN DD=0:CD=-B
820 IF B>0 THEN DD=B:CD=0
830 IF P=0 THEN LOCATE 7,Y:PRINT "Bal. C
/D":LOCATE 20,Y:PRINT USING C$;DD;:PRINT
" ";:PRINT USING C$;CD:Y=Y+1
840 IF P=8 THEN PRINT #8,TAB(7) "Bal. C/
D";:PRINT #8,TAB(20) USING C$;DD;:PRINT
#8," ";:PRINT #8, USING C$;CD
850 PEN 3:PRINT #P,TAB(20)"=============
===="
860 IF B=0 THEN DD=D:CD=C
870 IF B<0 THEN DD=D:CD=D
880 IF B>0 THEN DD=C:CD=C
890 IF P=0 THEN LOCATE 20,Y+1:PRINT USIN
G C$;DD;:PRINT " ";:PRINT USING C$;CD
900 IF P=8 THEN PRINT #8,TAB(20) USING C
$;DD;:PRINT #8," ";:PRINT #8, USING C$;C
D
910 IF A=4 THEN P=0:GOTO 170
920 PEN 2:PRINT:PRINT:PRINT TAB(10)"<Spa
ce bar for Menu>"
930 A$=INKEY$:IF A$<>" " THEN 930
940 GOTO 170
950 CLS:IF N=80 THEN BORDER 6,9:LOCATE 5
,11:PRINT"No more accounts can be opened
":GOSUB 1960:BORDER 1:GOTO 170
960 PRINT:PRINT:PRINT "What is the name
of the new account?"
```

```
970 PRINT:PRINT:INPUT;N$
980 N$=N$+"       ":N$=LEFT$(N$,7)
990 FF=0:FOR J=1 TO N
1000 IF N$=N$(J) THEN FF=1
1010 NEXT
1020 IF FF=1 THEN BORDER 6,9:PRINT:PRINT
 "Account in this name already open":GOS
UB 1960:BORDER 1:GOTO 170
1030 PEN 1:PRINT:PRINT:PRINT "Account to
 be opened, named:"
1040 PRINT:PRINT TAB(15)N$
1050 IF N>69 THEN BORDER 6,9:PRINT:PRINT
:PRINT "Only ";80-N;"more accounts allow
ed"
1060 PEN 2:PRINT:PRINT:PRINT "OK? (Y/N)"
1070 A$=UPPER$(INKEY$)
1080 IF A$<>"Y" AND A$<>"N" THEN 1070
1090 IF A$="Y" THEN N=N+1:N$(N)=N$:BORDE
R 1
1100 GOTO 170
1110 NC=-1:WHILE NC<0 OR NC>N
1120 PEN 3:LOCATE 1,21:INPUT "Credited a
/c no. ",NC$
1130 NC=VAL(NC$):WEND
1140 IF NC=0 THEN 950
1150 ND=-1:WHILE ND<0 OR ND>N
1160 LOCATE 1,22:INPUT "Debited a/c no.
",ND$
1170 ND=VAL(ND$):WEND
1180 IF ND=0 THEN 950
1190 D$="":WHILE LEN(D$)<>4 OR DA<1 OR D
A>31 OR MO<1 OR MO>12
1200 LOCATE 1,23:INPUT "Date (DDMM)";D$

1210 DA$=LEFT$(D$,2):MO$=RIGHT$(D$,2)
1220 DA=VAL(DA$):MO=VAL(MO$)
1230 WEND
1240 NK=0:AM$="":LOCATE 1,24:PRINT "Amou
nts (Pounds.Pence)?  ";
1250 A$=INKEY$:IF A$="" THEN 1250
1260 AA=ASC(A$):IF AA<46 OR AA>57 OR AA=
47 THEN 1250
1270 IF NK=5 AND A$<>"." THEN 1250
1280 PRINT A$;:IF A$="." THEN 1300
1290 AM$=AM$+A$:NK=NK+1:IF NK<6 THEN 125
0
```

```
1300 IF VAL(AM$)>65535 THEN 1240
1310 NK=0:PE$=""
1320 A$=INKEY$:IF A$="" THEN 1320
1330 AA=ASC(A$):IF AA<46 OR AA>57 THEN 1
320
1340 PRINT A$;:PE$=PE$+A$:NK=NK+1:IF NK<
2 THEN 1320
1350 AM=VAL(AM$):PE=VAL(PE$)
1360 CLS
1370 PEN 1:PRINT:PRINT:PRINT "The transa
ction to be entered is:"
1380 PRINT:PRINT:PRINT "Credit account:
";:PEN 3:PRINT N$(NC)
1390 PEN 1:PRINT:PRINT:PRINT "Debit acco
unt:  ";:PEN 3:PRINT N$(ND)
1400 PEN 1:PRINT:PRINT:PRINT "Date:
      ";:PEN 3:PRINT D$
1410 PEN 1:PRINT:PRINT:PRINT "Amount:
      ";:PEN 3:PRINT"# "AM$"."PE$
1420 PEN 2:PRINT:PRINT:PRINT:PRINT:PRINT
 "Is this correct? (Y/N)  ";
1430 A$=UPPER$(INKEY$)
1440 IF A$<>"Y" AND A$<>"N" THEN 1430
1450 IF A$="N" THEN 340
1460 U=20000+7*NR:NR=NR+1
1470 POKE(U),NC:POKE(U+1),ND:POKE(U+2),D
A:POKE(U+3),MO
1480 POKE(U+4),INT(AM/256):POKE(U+5),AM-
256*PEEK(U+4):POKE(U+6),PE
1490 IF NR>2990 THEN CLS:BORDER 6,9:LOCA
TE 3,11:PRINT "Only "3000-NR" transactio
ns left":GOSUB 1960:BORDER 1
1500 GOTO 170
1510 CLS:PEN 1:PRINT:PRINT:PRINT:PRINT "
Trial balance"
1520 month=-1:WHILE month<0 OR month>13
1530 PEN 2:PRINT:PRINT:INPUT "To the end
 of which month? (1-12)",month$
1540 month=VAL(month$):WEND
1550 EN=20003:WHILE PEEK(EN)<=month AND
EN<>20003+NR*7
1560 EN=EN+7
1570 WEND
1580 EN=EN-10:trial=0:Y=5
1590 PEN 1:CLS:IF A=6 THEN PRINT:PRINT:P
RINT:PRINT "Printing trial balance":P=8
```

```
1600 PRINT #P,"Trial balance to end of m
onth";STR$(month):PRINT #P
1610 IF P=0 THEN PRINT "A/c   Name
    Debit    Credit":PRINT
1620 IF P=8 THEN PRINT #8,"A/c   Name
       Debit    Credit":PRINT #8
1630 FOR J=1 TO N:B=0
1640 FOR K=20000 TO EN STEP 7
1650 IF PEEK(K)=J THEN B=B-PEEK(K+4)*256
-PEEK(K+5)-PEEK(K+6)/100
1660 NEXT
1670 FOR K=20001 TO EN+1 STEP 7
1680 IF PEEK(K)=J THEN B=B+PEEK(K+3)*256
+PEEK(K+4)+PEEK(K+5)/100
1690 NEXT
1700 IF INT(ABS(B)*100)=0 THEN 1760
1710 IF B>0 THEN DD=B:CD=0
1720 IF B<0 THEN DD=0:CD=-B
1730 IF P=0 THEN PEN 3:PRINT J:LOCATE 7,
Y:PRINT N$(J):LOCATE 20,Y:PRINT USING C$
;DD;:PRINT" ";:PRINT USING C$;CD
1740 IF P=8 THEN PRINT #8,J;:PRINT #8,TA
B(7)N$(J);:PRINT #8,TAB(20) USING C$;DD;
:PRINT #8," ";:PRINT #8,USING C$;CD
1750 trial=trial+DD:Y=Y+1
1760 NEXT
1770 PEN 3:PRINT #P,TAB(20)"===========
=====" 
1780 IF P=0 THEN  LOCATE 20,Y+1:PRINT US
ING C$;trial;:PRINT " ";:PRINT USING C$;
trial
1790 IF P=8 THEN PRINT #8,TAB(20) USING
C$;trial;:PRINT #8," ";:PRINT #8,USING C
$;trial
1800 IF A=6 THEN P=0:GOTO 170
1810 PEN 2:PRINT:PRINT:PRINT TAB(8)"<Spa
ce Bar for menu>"
1820 A$=INKEY$:IF A$<>" " THEN 1820
1830 GOTO 170
1840 CLS:PRINT:PRINT:PRINT:PRINT "Finish
ing"
1850 OPENOUT "NAMES"
1860 PRINT #9,N:PRINT #9,NR
1870 FOR J=1 TO N:PRINT #9,N$(J):NEXT
1880 CLOSEOUT
```

```
1890 SAVE "AMOUNTS",B,20000,21000
1900 PRINT "Backup? (Y/N) ";
1910 A$=UPPER$(INKEY$):IF A$<>"Y" AND A$
<>"N" THEN 1910
1920 IF A$="Y" THEN 1850
1930 PRINT "ACCOUNT finished"
1940 END
1950 FOR J=1 TO 3000:NEXT
1960 RETURN
```

9

Phone Call Coster

Being able to see the cost mounting up as one talks is a sure deterrent to overspending on the telephone account. Since rates vary so widely according to distance and time of day, it is hard for the average person to estimate what the cost of a call is likely to be. The program caters for calls of all distances, both Inland and International, including those dialled directly and those obtained with the assistance of the operator. While you are talking, a marker moves across the bottom of the screen indicating what fraction of the current charging period has elapsed. This lets you know when you should begin to try to close the conversation, rather than incur the extra expense of another whole period.

The charges, durations and rate of VAT shown in the listing are those in force at the time of writing. These are incorporated in DATA statements which are easy to revise when changes in rates are announced.

How to use the program

The operation of the program is as simple as the telephone call tariff will allow it to be. You are given full screen displays of charges and the more important regulations affecting them. Whenever you have to key in a letter, the prompting message indicates which letters are appropriate. There is no need to key ENTER at any stage. All these features make for quick and easy use of the program.

When you RUN the program, you are first asked to choose between calls made by direct dialling and those which are made through the operator. The next choice is between Inland calls (which includes the Channel Islands and the Irish Republic) and International calls.

When you have made your choice the screen displays a table of charges for calls of the kind you have selected. You now have to answer simple queries about the kind of call you wish to make.

Inland calls are of five kinds, depending mainly on distance. Each is represented by a 'Charge Letter'. You can find out the letter to use by consulting your local Telephone Dialling Codes booklet. The letter is printed to the left of the name of each exchange. Note that instead of 'b1' for low cost calls, this program uses the letter 'C'.

The time of day or night for which the different Charge Rates are in force (Cheap, Standard, and Peak) are displayed on the screen. The rate for a call depends on the time the call is begun. The timings may vary at holiday periods, as announced in the Press. Frequently the Cheap Rate is in operation over the whole of the Christmas break, for example.

Calls which are directly dialled are charged by the number of 'Charge units' used. There is a fixed charge for each unit, but the duration of the unit varies with distance and Charge Rate. For a Local call at the Cheap rate, the charge unit lasts three minutes, while for a call over 56 km during the Peak period a charge unit lasts only twelve seconds. Calls made by the operator are charged in a different way. Here the charge is by the minute, and parts of a minute are charged as a whole minute. There is a minimum charge equivalent to the charge for three minutes (nine minutes for local calls at the Cheap rate). On the whole, operator calls are more expensive than direct dialled calls, though there is a special 'Lower Charge' rate for calls that have to be connected by the operator because the caller cannot dial or because a dialled call has failed. If you are timing an Inland call by operator, the program asks you to say whether the normal or lower charge rate applies.

The charges for International calls depend on the Charge Band, which is given in the International section of the Telephone Dialling Codes booklet. In this program we do not distinguish between Band 5A and Band 5B. The difference is that there is no Cheap Rate for Band 5B, so remember to enter only 'S' (Standard) when making calls in this Band. As with Inland calls, International calls are charged on Charge Unit basis. The length of a charge unit may be very short – only 3.1 seconds on Band 5B! Calls made by the operator are charged by the minute, with a minimum charge for three minutes, just as with Inland calls. Calls to certain countries can only be made as operator calls.

When a call is made by the operator, it is possible to make a Personal Call, the timing of which does not begin until the person named is actually present at the telephone. There is a surcharge for this service, and the program asks if you are making such a call. For a few countries, all calls are regarded as personal calls, and the surcharge must always be added to the charge.

The program uses colour coding so that you can be sure that your call is being charged by the computer at the correct rate. All charge tables and displays for Inland calls are on a green screen, while those for International calls are on a blue screen. When you are making a dialled call (Inland or International) the text is red. When the call is dialled by the operator, the text is in white.

When you have keyed in the details of the call you are intending to make, a message asks you to key 'S' (for start) as soon as you are connected. In the case of a personal call, wait until your caller has answered in person. As soon as you press 'S', the screen display changes to show the duration of the call, its cost so far, and its cost so far with VAT added. You may notice that the charges including VAT are sometimes 1p more than those quoted in British Telecom's Telephone Charges guide. In the guide the VAT inclusive charges are rounded to the nearest penny. On the telephone bill, however, odd fractions are ignored, as is the usual commercial practice. The program ignores fractions of pence.

While the call is in progress an arrow moves across the bottom of the screen. This tells you what portion of your current charge period has elapsed. You will see the charge increases suddenly whenever the arrow reaches the right-hand edge of the screen, for then a new period begins and another unit of charge has been incurred. With dialled long-distance calls the arrow moves all too rapidly! If you watch the arrow and can bring your call to a conclusion just before it reaches the right of the screen you can have the maximum of telephoning time for the minimum of cost. With operator calls the arrow takes three minutes to make its first journey across the screen (nine minutes in the case of local calls at Cheap Rate). After that it takes only one minute to cross the screen, as the charge is increased at intervals of one minute from then on.

When you finish the call, press key 'F'. The display is then held, allowing you to note down the charge. Then press the space-bar, to make the computer ready for costing the next call.

Keying in

There are many slight variations and exceptions in the telephone call tariff that make it hard to cover all its intricacies with a few standard subroutines. Consequently, sections of the program duplicate lines from other sections closely but not exactly. You can save yourself a lot of typing by using the micro's editing facility. For example, type line 70, then use the COPY key to produce lines 80 and 90, altering the few figures and letters which differ in each line. Having typed in lines 200 to 310, you will find that these can easily be copied and edited to provide lines 640 to 760. But remember to give the correct line numbers to the lines you create by copying!

Program design

20 Setting Mode.

30 Dimensioning arrays.

40 Initialise variable.

50 Clear screen.

60–110 Reading charge and duration DATA.

120 Sets screen and text colours to default colours (text yellow; screen blue).

130–180 Directing program to main types of call.

190 Text red; screen green.

200–320 Displaying and inputting details of Inland Direct Dialled calls.

330–410 Timing and displaying cost of all Direct Dialled calls.

420 Text red; screen blue.

430–590 Displaying and inputting details of International Direct Dialled calls.

600–620 Directing the program to two kinds of operator call.

630 Text white; screen green.

640–800 Displaying and inputting details of Inland operator calls.

810–890 Timing and displaying cost of all operator calls.

900 Text white; screen blue.

910–1080 Displaying and inputting details of International operator calls.

1090–1100 Subroutine for displaying line of a charge table (dialled calls).

1110–1240 Subroutine for inputting and analysing details of inland calls.

1250–1280 Subroutine waiting for call to begin.

1290–1440 Subroutine for timing and costing dialled calls.

1450–1460 Subroutine for printing line of a charge table (operator calls).

1470–1620 Subroutine for timing and costing operator calls.

1630–1670 DATA statements containing durations and charges etc.

The program

```
10 REM ** PHONE CALL COSTER **
20 MODE 1
30 DIM DL(5,3),DT(5,2),PL(5,6),PT(5)
40 R=0
50 CLS
60 READ C:READ V
70 FOR J=1 TO 5:FOR K=1 TO 3:READ DL(J,K
):NEXT:NEXT
80 FOR J=1 TO 5:FOR K=1 TO 2:READ DT(J,K
):NEXT:NEXT
90 FOR J=1 TO 5:FOR K=1 TO 6:READ PL(J,K
):NEXT:NEXT
100 FOR J=1 TO 5:READ PT(J):NEXT
110 READ S1,S2
120 CLS:INK 1,24:INK 0,1
130 PRINT:PRINT:PRINT "Call dialled or m
ade by operator? (D/O)"
140 A$=UPPER$(INKEY$):IF A$<>"D" AND A$<
>"O" THEN 140
150 IF A$="O" THEN 600
160 PRINT:PRINT "Call inland or internat
ional? (L/T)"
170 RE$=UPPER$(INKEY$):IF RE$<>"L" AND R
E$<>"T" THEN 170
180 IF RE$="T" THEN 420
190 CLS:INK 1,6:INK 0,22
200 PRINT "Inland direct dialled calls:"

210 PRINT:PRINT "Std. rate: 8-9 and 13-1
8, M-F"
220 PRINT "Peak rate: M-F"
230 PRINT "Cheap rate at all other times
"
240 PRINT:PRINT "Time (sec.) for 1 charg
e unit ("C"P)"
```

```
250 PRINT:PRINT:PRINT"Charge-letter  che
ap standard peak"
260 PRINT:PRINT "Local       L";:J=1:GOS
UB 1090:PRINT " "
270 PRINT "Up to 56 KM A";:J=2:GOSUB 109
0:PRINT " "
280 PRINT "Over 56 KM  B";:J=3:GOSUB 109
0:PRINT " "
290 PRINT "Low cost    C";:J=4:GOSUB 109
0:PRINT " "
300 PRINT "Irish Rep.  E";:J=5:GOSUB 109
0:PRINT " "
310 GOSUB 1110
320 GOSUB 1250
330 TIMER=INT(TIME/300)
340 CLS:PRINT:PRINT:PRINT "Length of cal
l =" "
350 PRINT:PRINT:PRINT "Cost of call ="
360 PRINT:PRINT:PRINT"Cost including VAT
 ="
370 PRINT:PRINT:PRINT "Key 'F' when you
have finished"
380 GOSUB 1290
390 LOCATE 7,20:PRINT "Space bar for nex
t call"
400 A$=UPPER$(INKEY$):IF A$<>" "THEN 400
410 IF A$=" " THEN 120
420 CLS:INK 1,6:INK 0,2
430 PRINT:PRINT:PRINT "International dir
ect dialled calls:"
440 PRINT:PRINT "Std. rate: 8-20, M-F"
450 PRINT "Cheap rate at all other times
"
460 PRINT "No cheap rate on charge band
5B"
470 PRINT:PRINT "Time (sec.) for charge
unit (";C;"P)"
480 PRINT:PRINT:PRINT "Charge band
cheap       standard"
490 FOR J=1 TO 5
500 PRINT TAB(5)J TAB(18)DT(J,1)TAB(31)D
T(J,2)
510 NEXT
520 PRINT:PRINT:PRINT "Charge band (1-5)
?"
```

```
530 A$=INKEY$
540 CL=VAL(A$):IF CL<1 OR CL>5 THEN 530
550 PRINT "Rate (C,S)?"
560 A$=UPPER$(INKEY$):IF A$<>"C" AND A$<
>"S" THEN 560
570 IF A$="C" THEN R=1
580 IF A$="S" THEN R=2
590 GOTO 320
600 PRINT:PRINT:PRINT "Call inland or in
ternational? (L/T)"
610 RE$=UPPER$(INKEY$):IF RE$<>"L" AND R
E$<>"T" THEN 610
620 IF RE$="T" THEN 900
630 CLS:INK 1,26:INK 0,9
640 PRINT "Inland operator connected cal
ls:"
650 PRINT:PRINT"Std. rate: 8-9 and 13-18
, M-F"
660 PRINT "Peak rate: 9-13, M-F"
670 PRINT "Cheap rate at all other times
"
680 PRINT "Charge per minute (minimum 3
min; 9 min on local cheap rate)."
690 PRINT:PRINT TAB(19)"Normal       low
er"
700 PRINT "Charge-letter    CH  ST  PK
CH  ST  PK"
710 PRINT "Local       L";:J=1:GOSUB 145
0:PRINT" "
720 PRINT "Up to 56 KM A";:J=2:GOSUB 145
0:PRINT" "
730 PRINT "Over 56 KM  B";:J=3:GOSUB 145
0:PRINT" "
740 PRINT "Low cost    C";:J=4:GOSUB 145
0:PRINT" "
750 PRINT "Irish Rep.  E";:J=5:GOSUB 145
0:PRINT" "
760 GOSUB 1110
770 PRINT "Normal or lower charge (N,L)?
"
780 A$=UPPER$(INKEY$):IF A$<>"N" AND A$<
>"L" THEN 780
790 IF A$="L" THEN R=R+3
800 GOSUB 1250
810 TIMER=INT(TIME/300)
```

```
820 CLS:PRINT:PRINT:PRINT "Length of cal
l ="
830 PRINT:PRINT:PRINT "Cost of call ="
840 PRINT:PRINT:PRINT"Cost including VAT
 ="
850 PRINT:PRINT:PRINT "Key 'F' when you
have finished"
860 GOSUB 1470
870 LOCATE 7,20:PRINT "Space bar for nex
t call"
880 A$=INKEY$:IF A$<>" " THEN 880
890 GOTO 120
900 CLS:INK 1,26:INK 0,2
910 PRINT:PRINT:PRINT "International ope
rator connected calls:"
920 PRINT:PRINT "Standard rate at all ti
mes"
930 PRINT "Personal call surcharge (1)"S
1"P or (2)"S2"Pmay be payable"
940 PRINT:PRINT:PRINT "Charge per minute
 (minimum 3 min.)"
950 PRINT:PRINT:PRINT "Charge-band (1-5)
?"
960 FOR J=1 TO 5
970 PRINT TAB(5)J TAB(23)PT(J)
980 NEXT
990 PRINT:PRINT "Charge-band (1-5)?"
1000 A$=INKEY$
1010 CL=VAL(A$):IF CL<1 OR CL>5 THEN 100
0
1020 PRINT "Personal call surcharge? (0,
1,2)"
1030 A$=INKEY$
1040 IF A$<>"0" AND A$<>"1" AND A$<>"2"
THEN 1030
1050 IF A$="0" THEN SC=0
1060 IF A$="1" THEN SC=S1
1070 IF A$="2" THEN SC=S2
1080 GOTO 800
1090 FOR K=1 TO 3:PRINT TAB(8+K*8);DL(J,
K);:NEXT
1100 RETURN
1110 PRINT:PRINT"Charge letter (L,A,B,C,
E)?"
1120 A$=UPPER$(INKEY$):IF A$<>"L" AND A$
```

```
<>"A" AND A$<>"B" AND A$<>"C" AND A$<>"E
" THEN 1120
1130 IF A$="L" THEN CL=1
1140 IF A$="A" THEN CL=2
1150 IF A$="B" THEN CL=3
1160 IF A$="B" THEN CL=3
1170 IF A$="C" THEN CL=4
1180 IF A$="E" THEN CL=5
1190 PRINT "Rate (C,S,P)?"
1200 A$=UPPER$(INKEY$):IF A$<>"C" AND A$
<>"S" AND A$<>"P" THEN 1200
1210 IF A$="C" THEN R=1
1220 IF A$="S" THEN R=2
1230 IF A$="P" THEN R=3
1240 RETURN
1250 PRINT:PRINT "Key 'S' when you are c
onnected"
1260 PRINT:PRINT "Key 'F' when you have
finished"
1270 A$=UPPER$(INKEY$):IF A$<>"S" THEN 1
270
1280 RETURN
1290 MK=1:ML=1
1300 T=INT((TIME/300)-TIMER)
1310 IF T=0 THEN PP=0:GOTO 1340
1320 IF RE$="T" THEN PP=T/DT(CL,R):GOTO
1340
1330 PP=T/DL(CL,R)
1340 UP=INT(PP)
1350 CH=INT((UP+1)*C)
1360 VA=INT(CH*(100+V)/100)
1370 LOCATE 23,3:PRINT T" secs"
1380 LOCATE 23,6:PRINT CH" P "
1390 LOCATE 23,9:PRINT VA" P "
1400 LOCATE ML,22:PRINT" ";
1410 MK=40*(PP-UP):IF MK<1 THEN MK=1
1415 LOCATE MK,22:PRINT CHR$(240)
1420 ML=MK
1430 A$=UPPER$(INKEY$):IF A$<>"F" THEN 1
300
1440 RETURN
1450 FOR K=1 TO 6:PRINT TAB(13+K*4)INT(P
L(J,K)+0.5);:NEXT
1460 RETURN
1470 MK=1:ML=1
```

```
1480 T=INT((TIME/300)-TIMER)
1490 PP=T/60
1500 IF T<180 AND (RE$="T" OR RE$="L" AN
D(CL>1 OR CL=1 AND R>1)) THEN UP=2:PP=PP
/3+UP:GOTO 1530
1510 IF T<540 AND CL=1 AND RE$="L" AND R
=1 THEN UP=8:PP=PP/9+UP:GOTO 1530
1520 UP=INT(PP)
1530 CH=INT((UP+1)*PL(CL,R)):IF RE$="T"
THEN CH=INT((UP+1)*PT(CL))+SC
1540 VA=INT(CH*(100+V)/100)
1550 LOCATE 23,3:PRINT INT(T/60)"   mins
"
1560 LOCATE 23,6:PRINT CH" P "
1570 LOCATE 23,9:PRINT VA" P "
1580 LOCATE ML,22:PRINT " "
1590 MK=40*(PP-UP):IF MK<1 THEN MK=1
1595 LOCATE MK,22:PRINT CHR$(240)
1600 ML=MK
1610 A$=UPPER$(INKEY$):IF A$<>"F" THEN 1
480
1620 RETURN
1630 DATA 4.4,15,480,120,90,144,45,30,48
,16,12,60,20,15,15,8,8
1640 DATA 9.25,7.4,9.25,7.5,6.15,4.95,5.
65,4.8,4.1,3.1
1650 DATA 2.6667,8,8,.5556,3,3,10,13,16,
2,6,9,14,24,29,6,17,22
1660 DATA 13,20,25,5,13,18,26,40,40,26,4
0,40
1670 DATA 62,62,80,95,146,200,400
```

Variations

Telephone charges are certain to change from time to time, and one may always hope for a welcome reduction in VAT. Provided that the structure of the charging does not alter drastically, the program can easily be modified for new charge rates. The DATA lines are made up as follows:

Line 1630: The cost of one charge unit for a direct dialled call (currently 4.4p), the VAT rate (15%), then the times in seconds allowed for 1 charge unit for Inland dialled calls (in order, row by row, as listed on the screen display).

Line 1640: Times in seconds allowed for 1 charge unit for International dialled calls.

Lines 1650 and 1660: Charges (pence per minute) for Inland operator calls.

Line 1670: Charges (pence per minute) for International operator calls, the personal call surcharge (Bands 1–3) and the personal call surcharge (Bands 4–5).

10

Info File

Once you have this program on tape, you will find it useful in so many ways. There are dozens of uses for it in the home, from keeping a catalogue of collections of music and video recordings, or indexing your favourite recipes, to using it as a diary, a telephone book, or a birthday present book. It can be used in the office for holding personnel records or customer records. In the shop, it can be used for holding an up-to-date price list of goods. In the workshop it can catalogue machine parts or component specifications. If your interest lies in research of any kind, use the program for indexing references to the literature of your subject. The Club Secretary could use it for keeping membership details, and a summary of the agendas of meetings.

How the program works

The information is stored in what is called a file. Here the file is a cassette tape with data recorded on it. Like the file in the office, the data it holds all relates to the same subject. You can have a file called 'Video', for example, and others called 'Birthdays', or 'Address'. Any of these files can be taken and used with the INFO FILE program, making it one of the most versatile programs in this book.

Within the file the information is separated into records. You may put several hundred records in a file, if you like. The records are equivalent to the sheets of paper held in a loose-leaf binder. Each holds a certain amount of information and each holds information of the same kind, depending on the purpose of the file. For example, in a file called 'Video tapes' there will be a record for each tape in your collection.

The information in each record is divided into a number of fields.

This program allows you to have up to four fields in each record. Each record in a videotape file would hold the same information about each video tape: its title and a summary of the story, or what the tape is about, a cast list or list of performers, a brief note on what you or your family think of it and perhaps technical details such as its running time, and the number of times it has been run.

The program allows any field to be up to eighty characters long, giving ample room for brief comments such as those in some of the items listed above. However, fields that do not need many characters can be allotted as few as you wish, thus saving memory space and allowing more records to be held in the file.

As another example of fields, consider a file used as a spare parts catalogue. The fields might be:

Part number or code
Specification
Price per unit part
Number of parts per box

There is a practical example of the program in use in the next section.

The program provides for the amending of information at any time, so data such as prices or address can readily be changed when necessary.

The program includes a 'Find' option which allows you to pick out and view any single record or set of records. You could, for example, ask it to display the names and addresses of all of your friends and relatives who live in the same town as you do, or all video recordings starring Michael Caine.

Using the program

The first message displayed on running the program is 'Data tape ready? (Y/N)'. If you have not used the program before for a given file, key 'N'; if you already have a data tape carrying the information you want to look at or add to, key 'Y'.

If you key 'N', you are then asked to enter the details of the file that must first be created. The first item is its name, such as 'FRIENDS', 'RECIPES', 'SOFTWARE'. Write the name on the label of the cassette, so that you will always remember its exact name. The records can have up to four fields, each of which must have a name, too. The fields of a record of a

'FRIENDS' file might be called 'NAME', 'ADDRESS', 'TELEPHONE NUMBER' and 'BIRTHDAY'. You are also asked to decide how many characters are to be allocated to each field. Each field can have between one and eighty characters. The more characters you allot to each field, the fewer records the file can hold. Make sure that you allow enough characters for names and addresses, for example, as it is not possible to alter the number of characters later. Remember that the characters include spaces between words. When you have finished giving your requirements, the computer calculates how many records the file will be able to hold, and displays this figure. If this is enough, key 'Y'. If not, key 'N' and you are asked to repeat the entering of field names and numbers of characters, perhaps cutting down on the number of characters to allow an increased number of records.

When all is settled, and you have keyed 'Y', there is a short pause and then you will see the message 'Press REC and PLAY then any key'. Put an empty C-10 or C-15 tape into the tape-recorder. This must be rewound to the beginning, but not to the non-magnetic leader. Press RECORD and PLAY on the tape-recorder, then press any key of the micro. After three to four minutes the new file is created on the tape. When the recorder stops, the message 'Backup?' appears. It is a very good idea to have a backup, so key 'Y' and repeat the recording operation with another new tape. After this, press 'N', which brings the program to an end. You now have a data tape (or two). Run the program again and this time answer 'Y' to the first question.

If you already had a data tape, or have just started one, you will have keyed 'Y' to the question, 'Data tape ready?'. You are then asked to type in the file name; copy this exactly from the name written on the cassette. Make sure the tape is rewound, and then press PLAY, then any key. When the recorder stops, the menu is displayed on the screen. There is so much data to handle in this program, that there may be pauses of 30 seconds or so, particularly during loading and saving.

The menu offers three options:

(1) *Enter record:* This is for adding information to your file. When you choose this option the screen displays the name of the first field and, in brackets after the name, the maximum number of characters allowed. Type in whatever information you want to go into this field. Typing in is slightly different from normal typing on the

computer. There is no cursor but, as you type, your entry appears on the next line of the screen. When you have typed to within five characters of the allowed number, a gentle 'beep' from the loudspeaker reminds you that you are running out of space in that field. When you have typed as many characters as are allowed, the computer refuses to accept any more. You will notice that the routine refuses to accept double quotes ("). This is because the string-handling routines of the computer would take the double quotes to be the end of the text in the field.

When you have finished entering the information in that field, press ENTER. The computer will now display the name of the second field. Carry on in this way until all fields have been filled in. The computer then returns you to the menu.

If you want to leave a field empty for the moment, just press ENTER without typing anything. If you make a mistake when typing information, press the DEL key. This does not delete characters as it normally does. Instead it prints a vertical line at the end of your incorrect entry, to show that it has been discarded. You can now retype the correct version, which appears on the line below. You can use the delete function as often as you want until the entry is correct.

(2) *Find record:* This is the option by which you can examine (and alter, if you wish) any record. When you select this option you are first asked which fields are to be searched to find the record you want. If you are not sure, or want all fields to be searched, key 'Y' in response to all queries. Otherwise key 'N' to those fields which are not relevant, for it takes longer to search all fields than just one or two. This option is a very powerful one, for it can find a piece of information in such a variety of ways. When you have decided which fields are to be included in the search, you are asked to type in 'what is to be found'. The computer accepts up to ten characters, and the typing in is just the same as making an entry in Option 1.

What you type depends on how you want the search to be made. Here are some examples to give you an idea of what to type:

(a) If you want to look through every record on the file, type nothing – just press ENTER.

(b) If you know exactly which record you want to see, type in the details fairly fully. For example, if you have selected the 'NAME' field for the search and want to see the record of Joseph Bloggs, type

'Joseph Bloggs' and (unless you have more than one person of that name in your file) the computer will find and display Joseph Bloggs' record and no other.
(c) If you are not sure how Joe spells his name (is it Bloggs or Blogs or Blogg? is it Joseph or Joe?) try typing 'Blog'. The computer will find all records for which one of the names begins with 'Blog', so you are bound to find the record you want.
(d) If you want to find all names beginning with a given letter, type 'B' for example, and you will then be shown all records with names beginning with that letter, from Babb to Bywaters.
(e) The program can just as easily find groups of characters which occur in the middle of words or phrases. For example, if you search an 'ADDRESS' field for 'W1X 4BD' it will pick out all records which have that Post Code in the address. Similarly, if an employee's code includes the letter 'M' for all male employees, searching the corresponding field for 'M' results in all records of male employees being displayed.

If there is no record which has the features you are searching for, the message 'Not found' appears and the border of the screen flashes red and green. After a few seconds, you are returned to the menu.

As each record is displayed, you are given three options:

C = continue
A = amend
Space bar = menu

If you key 'C', the search continues to display the next record which conforms to your request. When no more such records are to be found in the file, the message 'Search completed' is displayed. After a few seconds you are returned to the menu.

Keying 'A' gives you the chance to alter or delete the whole or part of a record. You are shown each field of the record in turn. You can then type one of the following:

(a) ENTER: this deletes the field completely, leaving it empty.
(b) /: the field is retained, unaltered. Then press ENTER.
(c) Type in the new information which is to replace the old. Note that the new information *replaces* the old information – it is not added to it. Then press ENTER.

If you delete all the fields in a record, the empty record is removed from the file, making room for a new record. All records following the deleted one are renumbered.

After you have amended a record, the search continues to find any further records conforming to your original description. This allows you to call up a set of records, or even all of the records, and amend each one in turn.

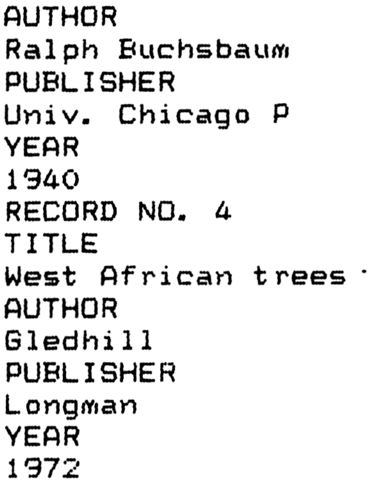
(3) *Finish:* It is essential not to finish the program by switching off the computer, or by pressing the ESCAPE key twice. Instead, choose Option 3. Rewind the data tape when the message 'Press REC and PLAY then any key' appears. Make sure it is positioned with the magnetic part against the tape head. Press RECORD and PLAY on the recorder, then press any key on the micro. When the data has been saved, you will then be asked if you want a backup. If so, repeat the operation, using your backup data tape. The program then finishes.

INFO FILE in action

Here is how INFO FILE can be used to hold the details of a personal library of books. The CAPS LOCK key was pressed before the program was run, so that we could use capitals and lower-case letters. But you can use all capitals if you prefer.

```
RECORD NO. 1
TITLE
Origin of species
AUTHOR
Charles Darwin
PUBLISHER
John Murray
YEAR
1859
RECORD NO. 2
TITLE
Life on Earth
AUTHOR
David Attenborough
PUBLISHER
Collins
YEAR
1979
RECORD NO. 3
TITLE
Animals with backbones
```

Fig. 10.1.



```
AUTHOR
Ralph Buchsbaum
PUBLISHER
Univ. Chicago P
YEAR
1940
RECORD NO. 4
TITLE
West African trees
AUTHOR
Gledhill
PUBLISHER
Longman
YEAR
1972
```

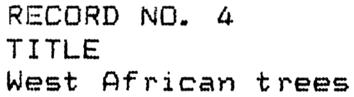
Fig. 10.1. (contd)

Figure 10.1 shows a printout of the first four records. As you can see the fields are:

TITLE of book
Name of AUTHOR
Name of PUBLISHER
DATE of publication

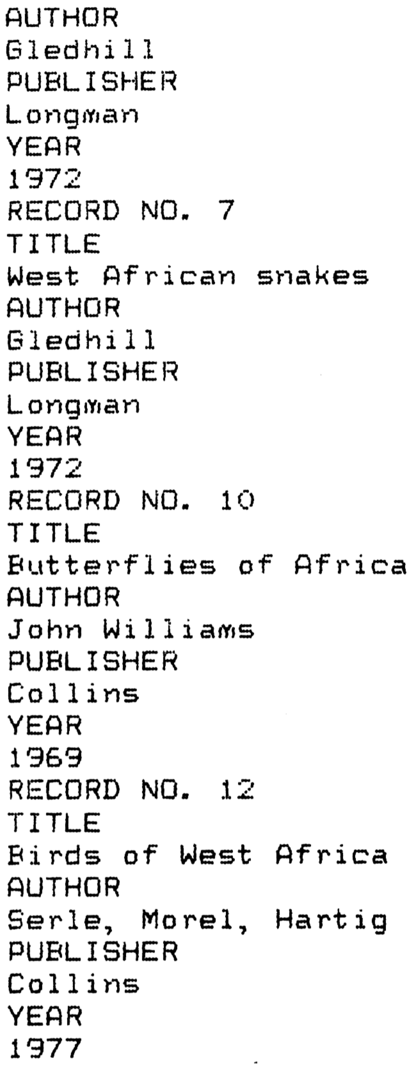
The records were set up with thirty characters for TITLE, twenty for AUTHOR and also for PUBLISHER, and four for DATE (only the year is normally required). This allowed more than 330 books to be listed.

Figure 10.2 shows the printout obtained when we asked INFO FILE to search for all titles relating to Africa. We keyed (2) to get to 'Find record'. Then we keyed 'Y' in response to 'Field 1 TITLE' and 'N' in response to the other fields. Finally we keyed 'Africa' when asked what was to be found. The books shown in Fig. 10.2 were listed on the screen, one at a time. Note that some titles have 'Africa' in them while others have 'African'. The search routine finds both sets of titles. If we had been interested only in West Africa, we could have keyed 'West Afric' (only ten characters allowed!). This would have given us records 4, 7 and 12, but not record 10.



```
RECORD NO. 4
TITLE
West African trees
```

Fig. 10.2.



```
AUTHOR
Gledhill
PUBLISHER
Longman
YEAR
1972
RECORD NO. 7
TITLE
West African snakes
AUTHOR
Gledhill
PUBLISHER
Longman
YEAR
1972
RECORD NO. 10
TITLE
Butterflies of Africa
AUTHOR
John Williams
PUBLISHER
Collins
YEAR
1969
RECORD NO. 12
TITLE
Birds of West Africa
AUTHOR
Serle, Morel, Hartig
PUBLISHER
Collins
YEAR
1977
```

Fig. 10.2. (contd)

Figure 10.3 shows the results of a search for 'Birds'. This gives us the West African book we obtained in Fig. 10.2, together with a book on the birds of New Guinea. Note that if some of the titles had included the word 'birds' (no capital B), or 'bird' (singular) they would not have been picked out in the search. Provided you do not have titles such as 'The third dimension' in your list, it would be better to key 'ird' when searching for bird books.

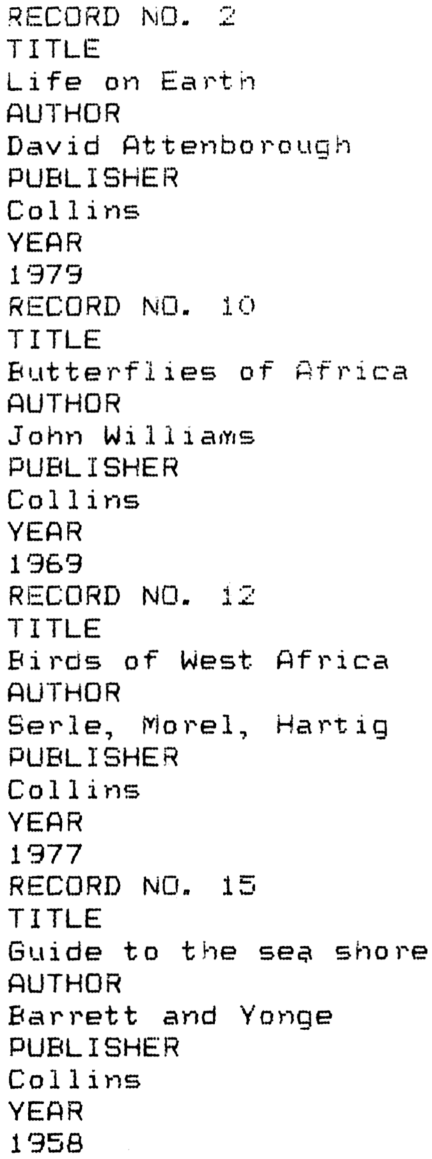
```
RECORD NO. 12
TITLE
Birds of West Africa
AUTHOR
Serle, Morel, Hartig
PUBLISHER
Collins
YEAR
1977
RECORD NO. 16
TITLE
Birds of New Guinea
AUTHOR
John Gould
PUBLISHER
Methuen
YEAR
1970
```

Fig. 10.3.

```
RECORD NO. 6
TITLE
Invest. in woodland ecology
AUTHOR
Prime
PUBLISHER
Heinemann
YEAR
1970
RECORD NO. 13
TITLE
The young botanist
AUTHOR
Prime
PUBLISHER
Nelson
YEAR
1963
```

Fig. 10.4. Result of searching for books written by a given author.

INFO FILE can also search the records looking for books from a given publisher. In Fig. 10.5 we see all the books published by Collins. Finally, in Fig. 10.6, we see the result of a search for all the books published in the nineteenth century. In this example, we keyed ‘Y’ in response to ‘Field 4 YEAR’, and then typed in ‘18’ as



```
RECORD NO. 2
TITLE
Life on Earth
AUTHOR
David Attenborough
PUBLISHER
Collins
YEAR
1979
RECORD NO. 10
TITLE
Butterflies of Africa
AUTHOR
John Williams
PUBLISHER
Collins
YEAR
1969
RECORD NO. 12
TITLE
Birds of West Africa
AUTHOR
Serle, Morel, Hartig
PUBLISHER
Collins
YEAR
1977
RECORD NO. 15
TITLE
Guide to the sea shore
AUTHOR
Barrett and Yonge
PUBLISHER
Collins
YEAR
1958
```

Fig. 10.5.

the search field. The search would have picked out books published in 1918, too, but there were no books of that vintage in the list.

```
RECORD NO. 1
TITLE
Origin of Soecies
AUTHOR
Charles Darwin
PUBLISHER
John Murray
YEAR
1859
RECORD NO. 9
TITLE
Natural History of Selbourne
AUTHOR
Gilbert White
PUBLISHER
Routledge
YEAR
1853
RECORD NO. 14
TITLE
The vegetable kingdom
AUTHOR
William Rhind
PUBLISHER
Blackie
YEAR
1868
```

Fig. 10.6.

Keying in

There are no special problems in keying in this program. Take care with the 'PRINT's in lines 510–550, as well as elsewhere. If you type the wrong number of these, the display will be spoilt.

Program design

20–80 Initialising.

90–380 Routine for setting up a new data tape (also used from line 270 for saving data).

390–500 Reading data from tape.
510–590 The menu.
600–700 Entering records.
710–970 Searching records.
980–1170 Amending records.
1180–1190 Finishing (completed in lines 270–380).
1200–1220 Subroutine to get key-press.
1230–1330 Subroutine used when typing into fields.
1340–1350 Subroutine to delay for five seconds.

The program

```
10 REM ** INFO FILE **
20 MODE 1
30 OPENOUT "AMSTRAD"
40 MEMORY HIMEM-1
50 CLOSEOUT
60 DIM N$(4),NC(4),FS(4),F$(4):LE=0:NC(4
)=10
70 E$=STRING$(80,"E"):S$=E$
80 INK 1,8:INK 2,20:INK 3,26
90 PRINT:PRINT:PRINT "Data tape ready? (
Y/N)"
100 GOSUB 1190
110 IF A$="Y" THEN 390
120 NN=0:TC=0:CLS:PRINT:PRINT "File name
?";:PEN 3:INPUT" ",N$
130 REM
140 PEN 2:LOCATE 1,4:PRINT "No. of field
s (1-4)?";:PEN 3:INPUT" ",NA$
150 N=VAL(NA$):IF N<1 OR N>4 THEN 120
160 FOR J=0 TO N-1
170 PEN 2:LOCATE 1,6+4*J:PRINT "Name of
field?"J+1;:PEN 3:INPUT" ",N$(J)
180 PEN 2:LOCATE 1,8+4*J:PRINT "No. of c
haracters (max 80)?";:PEN 3:INPUT" ",NC$
190 NC(J)=VAL(NC$):IF NC(J)<1 OR NC(J)>8
0 THEN 180
200 TC=TC+NC(J):NEXT
210 NR=INT(25000/TC)
220 PRINT:PRINT:PRINT "You can have up t
```

```
o"NR"records.":PEN 1:PRINT:PRINT "OK? (Y
/N)"
230 GOSUB 1190
240 IF A$="N" THEN 120
250 DIM D$(NR,N-1)
260 FOR J=1 TO NR:FOR K=0 TO N-1:D$(J,K)
="D":NEXT:NEXT
270 OPENOUT N$
280 PRINT #9,N:PRINT #9,NR:PRINT #9,NN
290 FOR J=0 TO N-1
300 PRINT #9,N$(J):PRINT #9,NC(J)
310 NEXT
320 FOR J=1 TO NR:FOR K=0 TO N-1:PRINT #
9,D$(J,K):NEXT:NEXT
330 PRINT#9,"X"
340 CLOSEOUT
350 PEN 1:PRINT "Back-up? (Y/N)"
360 GOSUB 1190
370 IF A$="Y" THEN 270
380 CLS:PRINT:PRINT:PRINT "INFOFILE fini
shed":END
390 PEN 1:PRINT:PRINT "File name?";:PEN
3:INPUT" ",N$
400 PRINT:PEN 1:OPENIN N$
410 INPUT #9,N:INPUT #9,NR:INPUT #9,NN
420 FOR J=0 TO N-1
430 INPUT #9,N$(J):INPUT #9,NC(J)
440 NEXT
450 DIM D$(NR,N-1)
460 FOR J=1 TO NR
470 FOR K=0 TO N-1
480 INPUT #9,D$(J,K)
490 NEXT:NEXT
500 CLOSEIN
510 CLS:PEN 2:PRINT:PRINT:PRINT "FILE: "
N$
520 PRINT:PRINT:PRINT TAB(5)"1)  Enter r
ecord"
530 PRINT:PRINT:PRINT TAB(5)"2)  Find re
cord"
540 PRINT:PRINT:PRINT TAB(5)"3)  Finish"
550 PEN 1:PRINT:PRINT:PRINT:PRINT "<  Se
lect by pressing numbered key  >"
560 A$=INKEY$
570 A=VAL(A$):IF A<1 OR A>3 THEN 560
```

```
580 IF A=2 AND NN=0 THEN 560
590 ON A GOTO 600,710,1170
600 NN=NN+1:CLS:PEN 2:PRINT:PRINT:PRINT
"FILE: "N$,TAB(25) "Record No."NN
610 FOR J=0 TO N-1
620 PEN 2:PRINT:PRINT "Field"J+1;:PEN 3:
PRINT N$(J);:PEN 2:PRINT " ("NC(J)")"
630 GOSUB 1220
640 D$(NN,J)=E$
650 PRINT"":NEXT
660 IF NN=NR THEN:BORDER 6,9:PEN 2:PRINT
:PRINT TAB(13)"**FILE FULL**"
670 PRINT:PRINT:PRINT "<Any key for more
 : Space bar for menu>"
680 A$=UPPER$(INKEY$):IF A$="" THEN 680
690 BORDER 1:IF A$<>" " THEN 600
700 GOTO 510
710 CLS:PEN 2:PRINT:PRINT:PRINT "FILE: "
;N$
720 FOR J=0 TO 3:FS(J)=0:NEXT
730 FOR J=0 TO N-1
740 PEN 1:PRINT:PRINT:PRINT "Search fiel
d ";J+1" ";:PEN 3:PRINT N$(J);:PEN 1:PRI
NT" Y/N ";
750 GOSUB 1190:PEN 3:PRINT " "A$
760 IF A$="Y" THEN FS(J)=1
770 NEXT
780 PEN 1:PRINT:PRINT:PRINT "Type what i
s to be found (MAX 10 chars)":PRINT:
790 FF=0:J=4:GOSUB 1220:S$=E$
800 L=1
810 J=0
820 IF FS(J)=0 THEN 850
830 X=INSTR(D$(L,J),S$)
840 IF X>0 THEN 900
850 J=J+1:IF J<N THEN 820
860 L=L+1:IF L<NN+1 THEN 810
870 IF FF=0 THEN:BORDER 6,9:PRINT:PRINT:
PRINT "Not found":GOTO 890
880 PRINT:PRINT TAB(12)"Search completed
"
890 GOSUB 1340:BORDER 1:GOTO 510
900 FF=1:CLS:PEN 2:PRINT:PRINT:PRINT "Re
cord No. "L:FOR M=0 TO N-1
910 PEN 2:PRINT:PRINT N$(M)
```

```
920 PEN 3:PRINT:PRINT D$(L,M)
930 PEN 3:PRINT:PRINT D$(L,M)
940 NEXT
950 PEN 1:PRINT:PRINT:PRINT "C=continue
: A=amend : Space bar=menu"
960 A$=UPPER$(INKEY$)
970 IF A$=" " THEN 520
980 IF A$="C" THEN 870
990 IF A$<>"A" THEN 960
1000 FOR J=0 TO N-1
1010 CLS:PEN 2:PRINT:PRINT:PRINT "Amend
record ";L
1020 PEN 1:PRINT:PRINT:PRINT "Field "J+1
" is:":PEN 3:PRINT:PRINT D$(L,J)
1030 PEN 1:PRINT:PRINT "Key / or the new
 version: ":PEN 3:PRINT:
1040 GOSUB 1230:IF E$<>"/" THEN D$(L,J)=
E$
1050 NEXT
1060 FD=0:FOR P=0 TO N-1
1070 IF D$(L,P)<>"" THEN FD=1
1080 NEXT:PRINT:PRINT:PRINT "Record amen
ded"
1090 GOSUB 1340
1100 IF FD=1 THEN 870
1110 IF L=NN THEN 1140
1120 FOR P=L TO NN-1
1130 FOR Q=0 TO N-1:D$(P,Q)=D$(P+1,Q):NE
XT:NEXT
1140 FOR Q=0 TO N-1:D$(NN,Q)=STRING$(NC(
Q),"D")
1150 NN=NN-1
1160 IF L>NN THEN 890
1170 GOTO 820
1180 CLS:PEN 2:PRINT:PRINT:PRINT "Finish
ing"
1190 GOTO 280
1200 A$=UPPER$(INKEY$)
1210 IF A$<>"Y" AND A$<>"N" THEN 1200
1220 RETURN
1230 E$=""
1240 A$=INKEY$:IF A$="" THEN 1240
1250 IF A$=CHR$(34) THEN 1240
1260 IF A$=CHR$(13) THEN RETURN
1270 IF A$=CHR$(127) THEN E$="":PRINT "¦
```

```
":GOTO 1230
1280 PEN 3:PRINT A$;
1290 E$=E$+A$
1300 IF LEN(E$)=NC(J) THEN RETURN
1310 IF LEN(E$)>NC(J)-5 THEN PRINT CHR$(
7);
1320 GOTO 1240
1330 RETURN
1340 FOR J=1 TO 2000:NEXT
1350 RETURN
```

11

Pools Punter

Fortunes have been won on the pools by people who just pick out the matches with a pin, or something equivalent. But it can be argued that you are more likely to find the winning combination by studying the past performance of the teams and using this information to predict the results of forthcoming matches. This program stores data on hundreds of past matches and uses it for pools prediction. It does all the hard work for you except the unavoidable task of keying in the data initially, though this is minimised by a special labour-saving and error-avoiding input routine.

The program can produce its predictions on scores only if you wish, but there is also scope for you to vary its method of working out the results, depending on which other factors you think are the most important in forecasting. It also lets you adjust the rating of individual teams to allow for the effects of such events as the recent loss of a star player.

When it has all been worked out, you may still like to have a bit of a gamble, so the program also provides a routine which simulates 'the pin', selecting 'lucky' matches purely at random.

This program assumes, of course, that the way a team has performed in the past is an indication of the way it will perform in the future. It cannot predict the unexpected. If you believe that you increase your chances of winning by studying past form, this program helps you do that more efficiently.

Using the program

Before you can use the program for prediction, you need to accumulate data referring to previous matches. Each week for a number of weeks you key in the results, each result consisting of four items of information: name of home team, name of away team, score of home team, score of away team. The program stores this data on a tape. One side of a C-15 tape holds the data for up to 7000 matches. If you follow the Australian teams too, it is better to use a separate tape for these, to reduce loading and saving time. Keying in the data can be done in short sessions if you wish. Week by week you can add the results of the latest matches and so gradually build up a reliable database from which the forecasts may be made.

When the program is run you are asked 'Data tape ready?'.

> (1) The first time you use the program, or when you are starting a new results tape, key 'N', and the program goes on to display the menu.
>
> (2) Once you have started using the program you will have a results tape carrying the data of previous matches which you have entered. Put this in the recorder, rewound to the beginning. Then key 'Y'. When you see the message 'Press REC and PLAY then any key' press the RECORD and PLAY buttons of the recorder. Then press any key on the micro. The file 'TEAMS' is loaded first. Then you are asked to press any key to load 'RESULTS'. It may take a few minutes to load the data into the computer, after which figures are displayed to show how many teams and matches are currently in memory. The maximum number of teams allowed is 200 and the maximum number of matches is 7000. If the figures displayed are near the maximum take care not to enter too many more teams or match results. Press the space-bar to see the menu.

(1) *Enter match results:* The way this option works depends on whether you have recently begun to use this program or whether you have been using it for a long time and have accumulated a large amount of data on the tape.

(a) When you are using this program for the first time, no team names will be stored in memory, so you will have to type the whole team name as you key in the results of each match. The computer begins by asking you to type in the first two letters of the name of the Home Team. If the Home team is Barnsley, for example, type in 'BA'. There is no need to press ENTER. The computer immediately looks through the list of team names which it may have already stored in its memory. As it does this it displays the names of all teams which have a name beginning with 'BA', together with the number given to that team on the computer's list. If there are no teams beginning with 'BA' no list will be displayed and you are then asked for the 'Team's full name'. Type this in – you are allowed up to ten letters. If it is longer, use a shortened form, such as 'HUDD'FIELD', as on the pools coupon. Press ENTER, when the name will be displayed and you are asked to type 'Y' if it is correct or 'N' if it is wrong. If you type 'N', you are asked to type the name again. If you type 'Y', the procedure is repeated for the Away Team. After this, you are asked to key in the number of goals scored by each team. Once again, you are given a chance to check that everything is entered correctly before the program continues.

If a list of teams is displayed it might show Barnet, Bangor and Barrow, but not Barnsley. In this case, key in the number '0' to indicate that the team you want is not listed. You will then be asked to type in its full name, as above.

If you have already typed in Barnsley on an earlier occasion, it appears on the list and all you have to do then is to type the number by which it is listed. So you never have to type in a team's name more than once!

(b) When you have been using the program for several weeks, you will have typed in most or all of the teams already. All you need to do when entering results is to type the first two letters of the name, and then its number on the list. This is much easier than having to type in the whole name, possibly getting it wrong. It avoids the possibility of typing errors and the risk of having the same team registered under three or four differently spelt names, such as 'MAN UTD', 'MAN UNITED' and 'MNCR U'.

Option 1 takes in the names and scores of both teams and stores them in memory. This data will follow after the data (if any) which has already been loaded from tape.

When you have finished entering the results, press the space-bar and you are returned to the menu to select another option. Incidentally, the program cannot be expected to predict anything unless it has data to work on. If you try to select options other than this one when no data has been loaded from tape or before any has been typed in, you will find that the program does not respond to your choice.

(2) *Set factors:* The program calculates a rating for all the teams based on one or more of the following factors:

The Home Team's wins when playing at home.
The Away Team's wins when playing away.
Either team's tendency to play a drawn match.
The Home Team's scores when playing at home.

The Away Team's scores when playing away.
The result of any previous match between the same two teams playing at the same venue.

You may think that certain of these factors are more important than others, and even that some should not be taken into account at all. If so, Option 2 gives you the chance to modify the prediction routine. If you do not want to use Option 2, then all factors are taken into account equally.

Select Option 2 and enter a number between 0 and 9 for each factor, when asked to do so. If you enter '0', that factor is ignored altogether in making predictions. A '9' gives a factor the maximum effect. You can return to this option after you have seen the predictions, and change the factors to see what effect this has when new predictions are made. The setting of factors is lost when the program is finished, so you need to set them afresh each time the program is run.

(3) *Adjust teams:* Quite often you may have information about some of the teams leading you to expect their performance to be better or worse than usual. There may be the transfer of a player, or perhaps they are simply in a bad patch at the moment. This option lets you feed this information into the prediction. You are asked the name of the team, as in Option 1, and then asked to key in a figure between −20 and +20. If you type '−20', the team's rating is reduced to 80% of its normal value before being used in the prediction. Similarly, typing in '20', increases its rating to 120%. Figures between −20 and 20 have proportionate effects. Team adjustments are lost when the program is finished, so should be made each time the program is run.

(4) *Predict results:* You must now provide the computer with the names of the teams which are to play against each other in the forthcoming matches. First you are asked 'Matches for prediction already entered?'. Unless you have already used this option on this occasion, key 'N'. You are then asked to enter the names of home and away teams, match by match. The easiest plan is to enter them in order, as on the current Pools Coupon. As in Option 1, you usually need key only the first two letters of the name, and then the number from the list displayed. If a team is playing which is not already registered, you type in its full name at this stage, and it will then be registered. Up to 100 matches may be keyed in for prediction at one time. When all matches have

been entered, press the space-bar. If there are many matches, prediction takes several minutes, possibly much longer. The program will not allow you to obtain a prediction when only a single match has been entered.

The first display shows the fifteen matches most likely to result in a home win. The matches are listed in order, the nearer the top of the list, the more likely the result is to be a home win. The values displayed on the right show you the relative ratings of each match. The higher the rating the greater the chance of a home win. If you want ten home wins, for example, take the first ten matches listed. However, if you fancy one of the matches lower down in the list, the ratings will give you an idea of how far below the 'top ten' your fancy lies. Pressing the space-bar causes the display to change to show the fifteen matches most likely to result in an away win. Once again, for the most likely away wins, take matches from the top of the list. Here the matches usually have negative ratings. The more negative the number, the more likely the result is to be an away win. Pressing the space-bar next takes you to a list of fifteen matches most likely to draw. The ratings range around zero, from small positive values to small negative values. Those nearest zero are most likely to be drawn matches. If you want eight draws, for example, pick eight matches from the centre of the list, including all those with zero rating if possible. Press the space-bar when you have looked at the draws list; you are then given the choice of pressing the space-bar to look through the lists again, or any key to return to the menu.

If you wish, you can go to Options 2 and 3 and then return to Option 4 for new predictions based on the changes you make. There is no need to enter the matches again; just key 'Y' when you are asked 'Matches for prediction already entered?'.

(5) *Lucky draw:* This option is for those who like to gamble. Possibly you might win more by using this option – who knows? Provided that ten or more matches have already been entered under Option 4, this option selects ten matches at random, and displays the details. You may find that some matches are picked twice, so there are not necessarily ten different matches on the list. If you want more matches, select this option a second time.

(6) *Finish:* When you leave the program, all matches, including the newly entered ones, must be saved to tape or disk. About 20 kilobytes of memory are set aside for storing data, which is held in a compact form. This means that the results of over 7000 matches can be held at one time and used for predictions. It is essential not to finish the program simply by keying ESCAPE twice or switching off the

power. When you want to finish with the program for the time being, select Option 6. The message 'Finishing' appears on the screen, together with the usual 'Press REC and PLAY then any key'. Rewind the data tape to the beginning, then wind it on so that it is the magnetic tape, not the non-magnetic leader, which is against the tape head. If the tape counter is set to '000' when the tape is fully rewound, run the tape on until it shows '005'. Press RECORD and PLAY on the recorder. Then press any key on the micro. The complete list of team names and all match details will then be put on to tape ready for the next time. The file 'TEAMS' is saved first. Then press any key to save the next file, 'RESULTS'. When this is finished, you are asked 'Back-up?'. This gives you a chance to make a second recording of the data on another tape. It is a wise precaution to do this, especially if you have several seasons of matches on record. Key 'Y', when you will be asked to repeat the whole recording procedure as already described. Keying 'N' ends the program.

Keying in

This program has lines which act directly on memory. Mistakes in typing could cause the program to crash inexplicably when run. Take special care with lines 30, 50, 150, and 490.

Program design

20–60 Initialising.
70–150 Loading from data tape.
160–200 Displaying number of teams and matches loaded.
210–320 Displaying menu.
330–530 Enter match results.
540–780 Set factors.
790–910 Adjust teams.
920–1830 Predict results.
1840–1930 Lucky draw.
1940–2050 Finish; saving data to tape.
2060–2350 Subroutine for inputting names of teams.
2360–2380 Subroutine: Yes or No?
2390–2450 Two subroutines for reading scores from data stored in RAM.
2460–2520 Subroutine to display tables of predictions.

Points of interest

Data is stored in two ways. The list of the names of teams is an array, T$(), capable of holding up to 200 teams. The results of matches are stored directly in an area of memory specially reserved for this purpose. Line 30 protects memory from address 20000 upward. Results are stored as three consecutive bytes in this area of RAM:

Number of home team (1 to 200)
Number of away team (1 to 200)
Scores of both teams

The two scores are compacted into a single byte by multiplying the home score by 16, then adding the away score (line 490). They can be recovered by ANDing with 240 for the home score (dividing the result by 16), and with 15 for the away score (lines 2400 and 2410). The limitation in this method is that scores of more than 15 goals have to be registered as 15 (lines 470 and 480). This slight disadvantage has to be set against the great advantage of reducing memory requirements, tape length and loading and saving times by 25%.

After two or three years, your data tape may carry so much information that you do not wish to retain the earlier records. The simplest, though most time-consuming method is to start a new tape and type in the recent records again. However, it is not difficult to arrange that the earlier records are omitted when you save data back on to tape using Option 6. Decide how many of the early records you wish to drop and multiply this number by 3, the number of bytes per record. Add 20000 to this to get the address in RAM from which you want saving to begin. Load the program and alter the first value in line 2000 as described below, so that saving begins from the newly calculated address. For example, if you want to discard the first 250 matches, saving should start from 20000 + 750 = 20750. Change the '20000' in line 2000 to 20750. Reduce the '21000' by the same amount, to 20250. When you run the program, it operates in the normal way except that Option 6 saves from the 251st match onward. This temporary change in the program is only for the occasion on which you decide to drop the early records. Do not change your POOLS PUNTER program permanently, for you will want normal saving to operate the next time it is used.

The program

```
10 REM ** POOLS PUNTER **
20 MODE 1
30 MEMORY 19999
40 DEFINT E-G,M,T
50 DIM T$(200),A(200),M(100,3),F(6):T=0:
R=20000:NM=0
60 FOR J=1 TO 6:F(J)=5:NEXT:FOR J=1 TO 2
00:A(J)=100:NEXT
70 CLS:PRINT "Data tape ready? (Y/N)"
80 GOSUB 2360
90 IF A$="N" THEN 210
100 OPENIN "TEAMS"
110 INPUT #9,R
120 INPUT #9,T:IF T=0 THEN 140
130 FOR J=1 TO T:INPUT #9,T$(J):NEXT
140 CLOSEIN
150 LOAD "RESULTS",20000
160 CLS:PRINT "Just loaded:"
170 PRINT TAB(16)T;" TEAMS"
180 PRINT TAB(16)INT((R-20000)/3);" MATC
HES"
190 PRINT:PRINT:PRINT TAB(8) "<Space bar
 to continue>"
200 A$=INKEY$:IF A$<>" " THEN 200
210 CLS:PRINT:PRINT TAB(9)"**  POOLS PUN
TER  **"
220 PRINT TAB(13)"------------"
230 PRINT:PRINT TAB(8)"1) Enter match re
sults"
240 PRINT:PRINT TAB(8)"2) Set factors"
250 PRINT:PRINT TAB(8)"3) Adjust teams"
260 PRINT:PRINT TAB(8)"4) Predict result
s"
270 PRINT:PRINT TAB(8)"5) Lucky draw"
280 PRINT:PRINT TAB(8)"6) Finish"
290 PRINT:PRINT:PRINT TAB(3)"< Select by
 pressing numbered key >"
300 A$=INKEY$
310 A=VAL(A$):IF A<1 OR A>6 THEN 300
320 ON A GOTO 330,540,790,920,1840,1940
330 CLS:PRINT:PRINT "Entering match resu
lts"
```

```
340 V$="HOME":GOSUB 2060
350 TH=FT
360 V$="AWAY":GOSUB 2060
370 TA=FT
380 CLS:PRINT "  Match between "T$(TH)"
(HOME) and ":PRINT TAB(3)T$(TA)" (AWAY):
"
390 GH=-1:WHILE GH<0
400 PRINT:PRINT "  Goals scored by "T$(T
H):INPUT "  ";GH$
410 GH=VAL(GH$):WEND
420 GA=-1:WHILE GA<0
430 PRINT:PRINT "  Goals scored by "T$(T
A):INPUT "  ";GA$
440 GA=VAL(GA$):WEND
450 PRINT:PRINT:PRINT "  Scores OK? (Y/N
)"
460 GOSUB 2360:IF A$="N" THEN 380
470 FS=1:IF GH>15 THEN GH=15
480 IF GA>15 THEN GA=15
490 POKE R,TH:POKE R+1,TA:POKE R+2,GH*16
+GA:R=R+3
500 PRINT:PRINT:PRINT "<Any key for more
 : Space bar for menu>"
510 A$=INKEY$:IF A$="" THEN 510
520 IF A$<>" " THEN 330:GOTO 210
530 GOTO 210
540 CLS:PRINT:PRINT "Set factors":FOR J=
1 TO 6:F(J)=1:NEXT
550 PRINT:PRINT "Key 0 to 9 for each fac
tor"
560 PRINT:PRINT TAB(10)"Home wins ";
570 A$=INKEY$:IF A$="" THEN 570
580 F(1)=VAL(A$):PRINT TAB(29)F(1)
590 PRINT:PRINT TAB(10)"Away wins ";
600 A$=INKEY$:IF A$="" THEN 600
610 F(2)=VAL(A$):PRINT TAB(29)F(2)
620 PRINT:PRINT TAB(10)"Draws ";
630 A$=INKEY$:IF A$="" THEN 630
640 F(3)=VAL(A$):PRINT TAB(29)F(3)
650 PRINT:PRINT TAB(10)"Home scores ";
660 A$=INKEY$:IF A$="" THEN 660
670 F(4)=VAL(A$):PRINT TAB(29)F(4)
680 PRINT:PRINT TAB(10)"Away scores ";
690 A$=INKEY$:IF A$="" THEN 690
```

```
700 F(5)=VAL(A$):PRINT TAB(29)F(5)
710 PRINT:PRINT TAB(10)"Previous matches
 ";
720 A$=INKEY$:IF A$="" THEN 720
730 F(6)=VAL(A$):PRINT TAB(29)F(6)
740 PRINT:PRINT:PRINT:PRINT TAB(5)"All O
K (Y/N)"
750 GOSUB 2360
760 IF A$="Y" THEN 210
770 GOTO 540
780 FOR J=1 TO 200:A(J)=100:NEXT
790 CLS:PRINT:PRINT "Adjust teams"
800 V$="":GOSUB 2060
810 AD=-21:WHILE AD<-20 OR AD>20
820 PRINT "What adjustment for "T$(FT)"?
":PRINT: INPUT"-20% TO +20%  ";AD:WEND
830 A(FT)=100+AD
840 PRINT:PRINT:PRINT TAB(10) "<Space fo
r menu>"
850 A$=INKEY$:IF A$<>" " THEN 850
860 GOTO 210
870 FOR J=1 TO 128
880 FOR K=2 TO 3
890 M(J,K)=0
900 NEXT
910 NEXT
920 CLS:PRINT "Matches for prediction al
ready entered? (Y/N)"
930 GOSUB 2360
940 IF A$="Y" THEN 1050
950 NM=1
960 CLS:PRINT "Predict results ":PRINT "
Match No "NM:V$="HOME"
970 GOSUB 2060
980 M(NM,0)=FT
990 PRINT:PRINT:PRINT "Match No "NM:V$="
AWAY"
1000 GOSUB 2060
1010 M(NM,1)=FT
1020 PRINT:PRINT:PRINT "Any key for more
 : Space bar to predict"
1030 A$=INKEY$:IF A$="" THEN 1030
1040 IF A$<>" " THEN NM=NM+1:IF NM<101 T
HEN 960
1050 IF NM<2 THEN 1030
```

```
1060 CLS:PRINT:PRINT:PRINT "    This may
 take several minutes!"
1070 FOR J=1 TO NM:M(J,2)=0:M(J,3)=0:NEX
T
1080 RS=20000:FOR J=1 TO NM
1090 IF F(1)=0 THEN 1160
1100 MM=0:MR=0:FOR K=RS TO R-3 STEP 3
1110 IF M(J,0)<>PEEK(K) THEN 1140
1120 GOSUB 2390
1130 IF GH>GA THEN MR=MR+1
1140 NEXT:IF MM=0 THEN M(J,2)=M(J,2)+50*
F(1):GOTO 1160
1150 M(J,2)=M(J,2)+100*MR/MM*F(1)
1160 IF F(2)=0 THEN 1230
1170 MM=0:MR=0:FOR K=RS TO R-3 STEP 3
1180 IF M(J,1)<>PEEK(K+1) THEN 1210
1190 GOSUB 2390
1200 IF GH<GA THEN MR=MR+1
1210 NEXT:IF MM=0 THEN M(J,3)=M(J,3)+50*
F(2):GOTO 1230
1220 M(J,3)=M(J,3)+100*MR/MM*F(2)
1230 IF F(3)=0 THEN 1350
1240 MM=0:MR=0:FOR K=RS TO R-3 STEP 3
1250 IF M(J,0)<>PEEK(K) THEN 1280
1260 GOSUB 2390
1270 IF GH=GA THEN MR=MR+1
1280 NEXT:IF MM=0 THEN M(J,2)=M(J,2)+50*
F(2):GOTO 1300
1290 M(J,2)=M(J,2)+50*(1+(MM-MR)/MM)*F(3
)
1300 MM=0:MR=0:FOR K=RS TO R-3 STEP 3
1310 IF M(J,1)<>PEEK(K+1) THEN 1340
1320 GOSUB 2390
1330 IF GH=GA THEN MR=MR+1
1340 NEXT:IF MM=0 THEN M(J,3)=M(J,3)+50*
F(3)
1350 IF F(4)=0 THEN 1420
1360 TH=0:TA=0:FOR K=RS TO R-3 STEP 3
1370 IF M(J,0)<>PEEK(K) THEN 1400
1380 GOSUB 2430
1390 TH=TH+GH:TA=TA+GA
1400 NEXT:IF TA+TH=0 THEN M(J,2)=M(J,2)+
50*F(4):GOTO 1420
1410 M(J,2)=M(J,2)+TH/(TA+TH)*100*F(4)
1420 IF F(5)=0 THEN 1490
```

```
1430 TH=0:TA=0:FOR K=RS TO R-3 STEP 3
1440 IF M(J,1)<>PEEK(K+1) THEN 1470
1450 GOSUB 2430
1460 TH=TH+GH:TA=TA+GA
1470 NEXT:IF TA+TH=0 THEN M(J,2)=M(J,2)+
50*F(5):IF TA+TH=0 THEN M(J,3)=M(J,3)+50
*F(5):GOTO 1490
1480 M(J,3)=M(J,3)+TA/(TA+TH)*100*F(5)
1490 IF F(6)=0 THEN 1590
1500 TH=0:TA=0:FOR K=RS TO R-3 STEP 3
1510 IF M(J,0)<>PEEK(K) OR M(J,1)<>PEEK(
K+1) THEN 1550
1520 GOSUB 2430
1530 TH=TH+GH
1540 TA=TA+GA
1550 NEXT
1560 IF TA+TH=0 THEN M(J,2)=M(J,2)+50*F(
6):M(J,3)=M(J,3)+50*F(6):GOTO 1590
1570 M(J,2)=M(J,2)+TH/(TH+TA)*100*F(6)
1580 M(J,3)=M(J,3)+TA/(TH+TA)*100*F(6)
1590 NEXT
1600 FOR J=1 TO NM
1610 M(J,2)=M(J,2)/100*A(M(J,0))-M(J,3)/
100*A(M(J,1)):M(J,3)=J
1620 NEXT
1630 FOR K=1 TO NM-1
1640 FOR L=K+1 TO NM
1650 IF M(L,2)>M(K,2) THEN 1680
1660 E=M(L,2):M(L,2)=M(K,2):M(K,2)=E
1670 E=M(L,3):M(L,3)=M(K,3):M(K,3)=E
1680 NEXT:NEXT
1690 CLS:PRINT:PRINT:PRINT "HOME WINS:"
1700 X=NM:Y=NM-15:IF Y<1 THEN Y=1
1710 Z=-1:GOSUB 2460
1720 CLS:PRINT:PRINT "AWAY WINS:"
1730 X=1:Y=15:IF Y>NM THEN Y=NM
1740 Z=1:GOSUB 2460:J=1
1750 CLS:PRINT:PRINT:PRINT "DRAWS:"
1760 IF SGN(M(J,2))=-1 THEN J=J+1:IF J<=
NM THEN 1760
1770 X=J-7:IF X<1 THEN X=1
1780 Y=J+7:IF Y>NM THEN Y=NM
1790 Z=1:GOSUB 2460
1800 CLS:PRINT:PRINT:PRINT "Any key for
menu : Space bar for review"
```

```
1810 A$=INKEY$:IF A$="" THEN 1810
1820 IF A$<>" " THEN 210
1830 GOTO 1690
1840 IF NM<9 THEN CLS:PRINT:PRINT:PRINT
"Too few matches":FOR J=1 TO 5000:NEXT:G
OTO 210
1850 CLS:PRINT "Lucky draw:"
1860 PRINT "Match"TAB(11) "HOME"TAB(26)
"AWAY"
1870 FOR J=1 TO 10
1880 R=INT(RND(1)*NM)+1
1890 PRINT R TAB(11)T$(M(R,0))TAB(26)T$(
M(R,1))
1900 NEXT
1910 PRINT:PRINT:PRINT:PRINT TAB(10) "<
Space bar for menu >"
1920 A$=INKEY$:IF A$<>" " THEN 1920
1930 GOTO 210
1940 CLS:PRINT:PRINT:PRINT "Finishing"
1950 OPENOUT "TEAMS"
1960 PRINT #9,R
1970 PRINT #9,T
1980 FOR J=1 TO T:PRINT #9,T$(J):NEXT
1990 CLOSEOUT
2000 SAVE "RESULTS",B,20000,21000
2010 PRINT "Backup? (Y/N) ";
2020 A$=UPPER$(INKEY$):IF A$<>"Y" AND A$
<>"N" THEN 2020
2030 IF A$="Y" THEN 1940
2040 PRINT:PRINT:PRINT "POOLS PUNTER fin
ished"
2050 END
2060 PRINT:PRINT V$" Team - first two le
tters ";
2070 A$=UPPER$(INKEY$):IF A$="" THEN 207
0
2080 PRINT A$;
2090 B$=UPPER$(INKEY$):IF B$="" THEN 209
0
2100 PRINT B$:T$=A$+B$
2110 IF T=0 THEN FT=0:GOTO 2150
2120 FT=0:FOR J=1 TO T
2130 IF T$=LEFT$(T$(J),2) THEN PRINT:PRI
NT J;" ";T$(J):FT=1
2140 NEXT
```

```
2150 IF FT=0 THEN 2300
2160 FT=-1:WHILE FT<0 OR FT>T
2170 PRINT: INPUT "Key one of these Nos,
 or '0' ";T$
2180 FT=VAL(T$):WEND
2190 IF FT=0 THEN 2300
2200 T$=T$(FT)
2210 PRINT:PRINT "The name of the "V$" t
eam is:":PRINT:PRINT TAB(12)LEFT$(T$,10)
2220 LOCATE 25,19:PRINT "...OK? (Y/N)"
2230 GOSUB 2360
2240 IF A$="N" THEN 2300
2250 T$=LEFT$(T$,10):FF=0:FOR J=1 TO T:I
F T$=T$(J) THEN FF=J
2260 NEXT
2270 IF FF>0 THEN FT=FF:GOTO 2290
2280 IF FY=1 THEN T=T+1:T$(T)=LEFT$(T$,1
0):FT=T
2290 CLS:RETURN
2300 CLS:INPUT "Teams full name ";t$
2310 PRINT:PRINT "Name OK? (Y/N)"
2320 FY=0:GOSUB 2360
2330 IF A$="Y" THEN FY=1:GOTO 2210
2340 GOTO 2300
2350 RETURN
2360 A$=UPPER$(INKEY$)
2370 IF A$<>"Y" AND A$<>"N" THEN 2360
2380 RETURN
2390 MM=MM+1
2400 GH=(PEEK(K+2) AND 240)/16
2410 GA=PEEK(K+2) AND 15
2420 RETURN
2430 GH=(PEEK(K+2) AND 240)/16
2440 GA=PEEK(K+2) AND 15
2450 RETURN
2460 PRINT "Match HOME"TAB(18) "AWAY"TAB
(30) "Rating"
2470 FOR J=X TO Y STEP Z
2480 PRINT M(J,3)TAB(7)T$(M(M(J,3),0))TA
B(18)T$(M(M(J,3),1))TAB(30)M(J,2)
2490 NEXT
2500 PRINT:PRINT:PRINT TAB(9) "<Space ba
r to continue>"
2510 A$=INKEY$:IF A$<>" " THEN 2510
2520 RETURN
```

12

Trend Finder

Are things getting worse or better? Or are they, perhaps just about the same as ever they were? Is productivity improving? Are the rejection rates falling? Is circulation increasing? Are there more or fewer birds feeding at the bird table than there used to be? These and other questions of similar nature can be answered by this program. It detects trends in any given series of figures. You need at least ten figures in the series, but it is preferable to have at least twenty, and you can have up to a hundred. The more figures you have the more reliably the program will be able to find out if they show a trend.

How to use the program

The program works equally well with measurements, numbers or grades. The weekly production, in tonnes, of a chemical plant is an example of measurement data. The numbers of ice-creams sold daily is an example of a number series. To assess the popularity of a new LP, market researchers might question a number of people (say 100) in a series of surveys, asking them to grade the LP on a scale ranging from 0 to 10. The series of totals of the 100 gradings obtained from each survey could be used as data for this program. Any trend in the popularity of the LP could then be detected.

The program accepts negative values as well as positive ones. Also, there is no need for the entries to have been taken at regular intervals. A series of so-called 'weekly' figures might, in practice, have gaps when the factory closes over the holidays, for example. Maybe the data for one week was inexplicably lost. If data is missing from the series, simply pass to the next item; do not enter '0' or try to estimate a value to replace the missing one.

As soon as the program is run, you are asked to key in the data.

When you have finished entering the data (at least ten items), key '9999'. This informs the computer that all the data has been entered. The figure '9999' is not incorporated into the calculations. The figures are redisplayed so that you can check them. Confirm that they are all correct by keying 'Y'. The screen then shows the warning 'Computing may take a while'. Normally, the calculation takes about ten seconds, though will take longer if you have nearly 100 figures.

Finally the computer displays the result. For example, you might see:

> "There is a chance between 98% and 99% that there is a trend"

This means that the likelihood of the figures indicating a real trend is very strong. Conversely, there is only a 1% to 2% chance of the rise or fall of the values along the series being due to chance factors. Most people would accept the fact that there is a slight chance (only 1 in 50 to 1 in 100) that there is not a trend, and therefore consider that a trend has been demonstrated.

If the percentages quoted in the display are in the region of 50% to 70%, there is quite a strong possibility of there not being a trend. With such figures, you should not accept that a trend is present.

We have assumed that a trend must go up or down, but it is also possible that there can be a 'no-change' trend. If the figures 'hover' closely and regularly above and below a given value instead jumping about irregularly, it is evident that there is a strong trend toward stability. A trend of this kind is detected by the program just as easily as a steady upward or downward trend. So, if the program tells you that the chances of a trend are high, examine the series of figures to check if the trend is upward, downward, or level. It must be emphasised that the test is much more reliable when there are more than twenty figures in the series. The more the better, except that, in practice, a long series may have a number of 'upward' and 'downward' trends, which may cancel each other out. A series which shows a rising trend at the beginning, followed by a falling trend, will probably not produce any significant result in this test. If you have a very long series, it may be worthwhile dividing it into several sub-series and testing each sub-series separately.

Keying in

Take special care with lines 480 and 740–760. In lines 740–760 note that there are both commas and full-stops (decimal points).

Program design

20–30 Initialising.
40–90 Entering and verifying data.
100 Warning message.
110 Data to array S() for sorting.
120–290 Quicksort to sort data.
300–310 Finding the median value.
320–400 Scoring data depending upon whether it is above or below the median.
410–460 Counting number of runs and numbers above and below median.
470–490 Calculating the statistics.
500–700 Assessing probabilities and displaying the results.
710–730 Inviting another analysis.
740–760 Statistical look-up table.

Points of interest

Note for the more statistically-minded reader. This program uses the Wald-Wolfowitz runs test adapted for detecting trends. Data which equals the median is allocated to the 'above-median' or 'below-median' categories at random. The DATA statements contain the values of *z* associated with selected probability (one-tailed) levels.

The program

```
10 REM ** TREND FINDER **
20 MODE 1
30 DIM D(100),S(101),R(101),SS(10,2),N(2
)
40 A$="N":WHILE A$="N":NT=1:CLS:PRINT "K
ey in data":PRINT
```

```
50 WHILE NT<100 AND D(NT-1)<>9999:PRINT
NT;
60 INPUT"   ";D(NT):NT=NT+1:WEND
70 NT=NT-2:CLS:PRINT "DATA ENTERED IS:"
80 FOR J=1 TO NT:PRINT:PRINT D(J);:NEXT
90 PRINT:PRINT:INPUT "All OK? (Y/N) ";A$
:A$=UPPER$(A$):WEND
100 CLS:LOCATE 7,5:PRINT "Computing may
take a while"
110 FOR J=1 TO NT:S(J)=D(J):NEXT
120 M1=1
130 SS(1,1)=1:SS(1,2)=NT
140 M2=SS(M1,1):M3=SS(M1,2):M1=M1-1
150 M4=M2:M5=M3:M6=S(INT(RND(1)*(M3-M2)+
0.5)+M2)
160 IF S(M4)<M6 THEN M4=M4+1:GOTO 160
170 IF M6<S(M5) THEN M5=M5-1:GOTO 170
180 IF M4>M5 THEN 200
190 M7=S(M4):S(M4)=S(M5):S(M5)=M7:M9=R(M
4):R(M4)=R(M5):R(M5)=M9:M4=M4+1:M5=M5-1
200 IF M4<=M5 THEN 160
210 IF M5-M2>=M3-M4 THEN 250
220 IF M4>=M3 THEN 240
230 M1=M1+1:SS(M1,1)=M4:SS(M1,2)=M3
240 M3=M5:GOTO 280
250 IF M2>=M5 THEN 270
260 M1=M1+1:SS(M1,1)=M2:SS(M1,2)=M5
270 M2=M4
280 IF M2<M3 THEN 150
290 IF M1>0 THEN 140
300 IF NT/2=INT(NT/2) THEN MN=S(NT/2):IF
 S(NT/2+1)<>MN THEN 370
310 MN=S(NT/2+0.5)
320 FOR J=1 TO NT
330 IF D(J)<MN THEN R(J)=0
340 IF D(J)=MN THEN R(J)=INT(RND(1)+0.5)
350 IF D(J)>MN THEN R(J)=1
360 NEXT:GOTO 410
370 FOR J=1 TO NT
380 IF D(J)<=MN THEN R(J)=0
390 IF D(J)>MN THEN R(J)=1
400 NEXT
410 N(1)=0:U=0:R(NT+1)=R(NT)+1
420 FOR J=1 TO NT
430 IF R(J)=1 THEN N(1)=N(1)+1
```

```
440 IF R(J)<>R(J+1) THEN U=U+1
450 NEXT J
460 N(2)=NT-N(1)
470 UU=2*N(1)*N(2)/(N(1)+N(2))+1
480 S=SQR(2*N(1)*N(2)*(2*N(1)*N(2)-N(1)-
N(2))/(N(1)+N(2))^2/(N(1)+N(2)-1))
490 Z=(ABS(UU-U)-0.5)/S
500 IF Z<0 THEN Z=0.001
510 IF N(1)*N(2)<18 OR N(1)<5 OR N(2)<5
THEN CLS:LOCATE 13,7:PRINT "Not enough d
ata":GOTO 710
520 R=0
530 R=R+1:READ ZL
540 IF ZL=Z THEN 560
550 IF ZL>Z THEN 530
560 IF R<10 THEN P=R/10
570 IF R>9 AND R<19 THEN P=R-9
580 IF R>18 THEN P=10+5*(R-19)
590 IF ZL<Z THEN 640
600 CLS:LOCATE 8,5:PRINT "There is a"100
-P"% chance"
610 LOCATE 9,7:PRINT "that there is a tr
end"
620 GOTO 690
630 IF R<11 THEN PL=(R-1)/10
640 IF R>10 AND R<20 THEN PL=R-10
650 IF R>19 THEN PL=10+5*(R-20)
660 CLS:LOCATE 7,5:PRINT "There is a cha
nce between"
670 LOCATE 18,7:PRINT 100-P"%"
680 LOCATE 19,9:PRINT "and"
690 LOCATE 18,11:PRINT 100-PL"%""
700 LOCATE 10,13:PRINT "that there is tr
end"
710 LOCATE 5,23:PRINT "<Space bar for an
other analysis>"
720  A$=INKEY$:IF A$<>" " GOTO 720
730 RUN
740 DATA 3.085,2.88,2.75,2.65,2.575,2.51
,2.455,2.41,2.365
750 DATA 2.327,2.052,1.881,1.751,1.645,1
.555,1.476,1.405
760 DATA 1.341,1.282,1.037,.842,.675,.52
4,.385,.253,.126,0
```

13

Supersound

The *User Instructions* book of the CPC464 states that 'The SOUND feature of the CPC464 is one of the most complex extensions to BASIC ...'. Most users who have tried to write programs with sounds in them would agree with this. The computer has a versatile sound generating chip which is able to produce an exciting variety of sounds and sound effects. Complexity is unavoidable. But, although it is complex, it is not necessarily difficult to handle. All that is needed is a program, such as the one described in this chapter, to help you design the sounds you require.

One of the chief problems about producing sounds on any computer is that, given a SOUND statement with its parameters, most people find it very difficult to predict how it will sound when executed. They need to hear the sound, and then repeatedly alter one or more of the parameters and listen to the sound again. Much experimentation of this kind is required to develop a SOUND statement to produce the right effect. This experimentation will usually include trying numerous variations of the parameters in the volume and tone envelopes as well as those of the SOUND command itself. SUPERSOUND allows you to build up a set of definitions of sounds, volume envelopes (ENV) and tone envelopes (ENT), and to modify and test them until you obtain exactly what you require. When the sound or sounds are as you want them, the parameters of the SOUND and envelope statements may be read directly from the screen.

Using the program

It is assumed that you have read the section on sound in the Beginners' Foundation Course (pages F3.16 to F3.20) in the *User Instructions* book, and have also looked at Chapter 6, the Sound

Primer. Refer to these sections of the *User Instructions* book for explanations of terms used below.

When the program is run, the screen clears and a menu, or list of commands, is displayed at the bottom left. You will also see the word 'QUEUES' and the letters 'A', 'B' and 'C' at the bottom right. We will explain the purpose of these later.

The menu allows you to select one of five different routines by typing a number from 1 to 5:

1 = Play. This routine makes the computer play a note or sequence of notes.
2 = Compose. Use this routine to set up a sequence of notes on one or more channels, ready to be played.
3 = Define sound. This routine is for defining SOUND commands.
4 = Volume envelope. Define volume envelopes (ENV) by using this routine.
5 = Tone envelope. Define tone envelopes (ENT) with this routine.

If you require only simple notes or sound effects, there is no need to use routines 4 or 5. Until you are used to the way the SOUND command works, it is best to use only routines 1 to 3.

Defining sounds

We will start by describing this routine. Until you have defined some sounds, there is nothing to compose with and nothing to play! Press key '3' to enter this routine. As soon as you do so, a table of sound commands appears at the top left of screen. This has space for up to fifteen different sound commands. They are identified by letters A to O in the first column of the table. The other columns of the table are headed as follows (see *User Instructions* book section 6.10.1 for full explanations):

Ch	Channel status
Tone	Tone period
Dur	Duration
V	Volume
v	Volume envelope
t	Tone envelope
N	Noise period

The first time this table appears, all these columns are filled with zeros, but the zeros will be replaced by other values as you define each sound.

The command area of the screen at bottom left now displays the message 'Sound letter (A to O)?'. This allows you to select which sound you wish to define or re-define. Key the appropriate letter. You are now asked to key in some of the parameters:

Tone period
Duration
Initial volume

If you are intending to use a volume envelope with the sound, it is usually best to let the envelope take care of the volume. In this case, enter '0' as the initial volume.

These three parameters are all that are needed for the simpler sounds. You are next asked if you wish to specify any envelopes to control the sound, or if noise is to be added. The question which appears on the screen is 'Envelopes or noise (Y/N)?'. If you key 'Y', you are asked to key in the number of the volume envelope, the number of the tone envelope, and the noise period. Type '0' for any of these three that you do not wish to specify.

When you have finished specifying envelopes and noise, or if you typed 'N' when asked if you wished to specify them, the parameters that you have just keyed in appear in the SOUNDS table. You are asked 'Another sound (Y/N)?'. Typing 'Y' leads to the sequence above being repeated. You can define another sound or, if you have made a mistake in the sound you just defined, you can re-define it. If you type 'N', the command menu reappears.

Composing sound sequences

Having defined one or more sounds, you will want to know what they sound like. Your next task is to place the sound commands into the computer's sound channels. Select this option from the menu by keying '2'. The area at bottom right of the screen displays the contents of the sound channels. The display has the same layout as the diagram on page 4 of Chapter 6 of the *User Instructions* book. The top row of the display (labelled A) shows five '@' symbols. These indicate that the five places in the Channel A queue are empty at present. The other two channels, B and C, are empty too.

The command area displays the message 'Channel A (Y/N)?'. If you wish to put sound commands into this channel, type 'Y'. You are then asked 'Sound (@or A to O)?'. This is inviting you to key in the letter representing the first sound to be heard on Channel A. You are then asked if you wish this sound to 'rendezvous' with a sound on another channel. The action of the 'rendezvous' facility is explained later. To 'rendezvous' two sounds, type the letter of the channel which will contain the sound with which you want to 'rendezvous' ('B' or 'C', if you are setting Channel A). If you do not wish for a 'rendezvous', type '@'.

You will hear no sounds at this stage, but you will see the letter of the selected sound appear in the first place on the display of Channel A. It is important that you type a letter which corresponds to a sound that you have already defined or intend to define before playing. Trying to play an undefined sound upsets the working of the program! If you have specified a 'rendezvous', the letter of the 'rendezvous' channel appears on the bright yellow band beneath the sound letter.

The routine asks for a sound letter and a 'rendezvous' letter for all five places in the queue. If you do not wish to enter a sound in a given place, key '@' for both sound and 'rendezvous'.

The sequence above is repeated for the other two channels. Key 'N' if you do not wish to put sounds in these channels.

If you wish to synchronise two channels by using the 'rendezvous' facility, make sure that both the notes to be synchronised are given a 'rendezvous' letter. For example, suppose that you wish to synchronise a note in Channel A with a note in Channel C. When entering the note in Channel A, you must specify that it is to 'rendezvous' with C. When entering the other note in Channel C, you must specify that it is to 'rendezvous' with A. When the 'Play' command is being executed, one of these two notes will probably reach the head of its channel before the other. When it reaches the head, it will not be sounded. Instead, there is silence from that channel until the other one of the pair reaches the head of its channel. Then the two notes sound together.

After you have been given the option to enter notes in all three channels, you are returned to the command menu.

Playing

This is option 1, which is placed first on the menu because this is the one you will want to use most often. To play the note, simply press key '1'. The 'SOUNDS' table clears and is redisplayed. Before the first time you select 'Play', the 'Ch' (Channel status) column of this table holds only zeros. Now that the notes have been allocated to particular channels in order to be played, the 'Ch' column shows appropriate status values. If the 'rendezvous' facility has not been used, these are '1' for Channel A, '2' for Channel B and '4' for Channel C. If the 'rendezvous' facility has been used, the channel status value is increased by 8 ('rendezvous' with A), 16 (with B) or 32 (with C). If the same note is to be played several times, in different channels or with different 'rendezvous' status, it needs a different channel status value on each occasion. The channel status value shown in the table is that which it holds when played at its right-most and lowest position in the queue table.

While the table is being displayed, the note or sequence of notes is sounded. You may repeat the sequence as many times as you like by pressing '1' again.

Defining volume envelopes

Key '4' to select this option. The table of volume envelopes is displayed at the top right of the screen. This shows the parameters of the fifteen volume envelopes. The column headings correspond to the parameter names given in the *User Instructions* book, Chapter 6, page 9. P, Q and R refer respectively to step count, step size and pause time. The numerals 1 to 5 following these refer to the five sections of the envelope.

As soon as the table has been displayed, you are asked to key the number of the envelope to be defined. If you key '0', the routine ends and you are returned to the menu. Selecting option 4 and then keying '0' is a convenient way of obtaining a display of the volume envelopes for reference while defining sounds, or composing sound sequences. If you key a number 1 to 15, you are asked to enter the step count, the step size and the pause time for each of the sections 1 to 5, in turn. If you wish to define fewer than five sections, key '0' for the values in the unwanted sections. After all values have been entered, the values appear in the table. You are then asked 'Another envelope (Y/N)?'. Key 'Y' or 'N' either to repeat the sequence or to return to the menu.

Defining tone envelopes

This routine is similar to that for the volume envelopes. The table of tone envelopes is displayed at top right of the screen, replacing the table of volume envelopes, if this is already displayed there. Keying '0' as the envelope number returns you to the menu, leaving the tone envelope table on display. The column headings correspond to the parameter names given in the *User Instructions* book, Chapter 6, page 10. Headings T, V and W refer respectively to step count, step size and pause time. The numerals 1 to 5 refer to the five sections of the envelope. The way of entering the values is exactly as described for the volume envelope.

Designing sounds

A good way to begin is to use this program to produce the sounds described in the Foundation Course. Then modify these, altering the values of one parameter at a time, to see what effect each change produces.

When you have designed a sound which you would like to use in one of your own programs, just copy the parameter list for the SOUND, ENV and ENT commands directly from the screen. The values displayed there are exactly those that should be used in the BASIC program. The only exceptions may be:

(a) As explained above, if you have used the same sound several times, its 'Ch' value will be that referring to its last occurrence in the QUEUES table. Values required for earlier occurrences will be different if the sound is to be sent to other channels or has a different 'rendezvous' specification. The *User Instructions* book explains how to calculate channel status values.

(b) You may have specified a tone envelope with a negative envelope number, to make the tone envelope repeat for the duration of the sound. The envelope number is always shown as a positive number in the tone envelope table, but should be keyed in as a negative number when typing the corresponding ENT statement in the BASIC program.

(c) You may wish to 'hold' certain sounds, or to 'flush' a channel. If so, the channel status value must be increased by 64 or 128, as explained in the *User Instructions* book, Chapter 6, page 6.

Keying in

This program divides the screen into four windows:

1. SOUNDS table
2. Volume or tone envelopes table
3. Commands menu
4. QUEUES table

It is essential that PRINT, INPUT, CLS and LOCATE instructions are sent to the correct windows. These are indicated by the '#1', '#2', '#3' or '#4' typed after these keywords. Take care to type the correct number in each case, since using the wrong number will completely spoil the display. The '#' (hash) is not required in lines 60 and 70, as the PRINT and LOCATE statements refer to window 0, i.e. to the whole screen. Typing mistakes in the lines which fill the tables will cause chaos. In particular, adhere precisely to the spacing in lines such as 890 and 900. A single space too few or a space too many will completely ruin the whole table.

Take care to include all the semicolons (;) in PRINT statements, especially in lines such as 1060, 1120 to 1150 and in several other lines.

Points of interest

When string R$ (defined in line 40) is used, it causes PAPER and PEN colours to be exchanged. This makes it easy to print the columns of the tables alternately in yellow-on-blue and in blue-on-yellow. Line 1060 is an example of a line on which R$ is used. Line 1060 also uses Z3$, Z4$ and Z6$, which are defined on line 40 as strings of hashes. Defining such strings is a convenient way of setting up PRINT USING statements without having to key in the actual string of hashes each time.

Program design

20–50 Initialise by setting Mode 2, defining integers (all letters of the alphabet except A and Z, which are used for string variables or constants), arrays, string constants, and windows.

60–70 Display labels for windows 3 and 4.

80–120 Selection of options from menu, in window 3.

130–220 The 'Play' routine. Lines 160 to 180 calculate the channel status. Line 190 sends the SOUND parameters (from array S(sound no., parameter)) to the sound channels.

230–400 The 'Compose' routine. Details are stored in a three-dimensional array, Q(channel, position, sound number or 'rendez-vous' status).

410–580 The 'Define sounds' routine. In line 470 the lower limit of duration is set to −32767, even though the SOUND command can actually accept −32768. The reason for this is explained below.

590–720 The volume envelope routine. Details are held in columns 16 to 30 of array E(envelope number, column) and transferred to the computer's memory at line 670.

730–860 The tone envelope routine. Details are held in columns 1 to 15 of array E and transferred to the computer's memory at line 810.

Column 0 of the array holds −1, 0 or 1, depending on the sign of the tone envelope number.

870–930 Subroutine for displaying volume or tone envelope table in window 2. If VT=0, it displays the tone envelopes; if VT=15, it displays the volume envelopes.

940–960 Subroutine for displaying one line of a volume or tone envelope table.

970–1010 Subroutine for displaying a message and accepting an integer input, within a given range (LL to UL). Since the WHILE...WEND loop requires M to begin with an out-of-range value, and since the duration parameter of the SOUND statement encompasses the entire range allowed to integers (−32768 to +32767), we have had to make −32768 unacceptable as a value for duration.

1020–1050 Subroutine for displaying the SOUND table in window 1.

1060–1090 Subroutine for displaying one line of the SOUND table.

1100–1160 Subroutine for displaying the QUEUES table in window 4.

The program

```
10 REM ** SUPERSOUND **
20 MODE 2:DEFINT B-Y
30 DIM E(15,30),Q(3,5,1),S(15,7)
40 R$=CHR$(24):Z2$="##":Z3$="###":Z4$="#
###":Z6$="######"
50 WINDOW #1,1,22,1,17:WINDOW #2,24,80,1
,17:WINDOW #3,5,40,20,25:WINDOW #4,52,61
,20,25
60 LOCATE 5,19:PRINT "COMMANDS":LOCATE 5
2,19:PRINT "QUEUES":LOCATE 68,19
70 LOCATE 50,20:PRINT "A":LOCATE 50,22:P
RINT "B":LOCATE 50,24:PRINT "C"
80 CLS #3:PRINT #3, "1 = Play":PRINT #3,
"2 = Compose":PRINT #3,"3 = Define sound
"
90 PRINT #3, "4 = Volume envelope":PRINT
 #3, "5 = Tone envelope"
100 PRINT #3,"Key 1 to 5"
110 M=VAL(INKEY$):IF M<1 OR M>5 THEN 110
```

```
120 ON M GOTO 130,230,410,590,730
130 CLS #3:PRINT #3,"SOUNDING"
140 FOR N=1 TO 5:FOR Q=1 TO 3
150 P=Q(Q,N,0):IF P=0 THEN 200
160 CS=Q:IF CS=3 THEN CS=4
170 IF Q(Q,N,1)>0 THEN CS=CS+2^(Q(Q,N,1)
+2)
180 S(P,1)=CS
190 SOUND CS,S(P,2),S(P,3),S(P,4),S(P,5)
,S(P,6),S(P,7)
200 NEXT:NEXT
210 GOSUB 1020
220 GOTO 80
230 GOSUB 1100
240 CLS #3:FOR Q=1 TO 3
250 PRINT #3,"Channel "CHR$(Q+64)" (Y/N)
";:INPUT #3,AQ$
260 IF UPPER$(AQ$)="N" THEN 390
270 FOR N=1 TO 5
280 CLS #3:INPUT #3,"Sound (@ or A to O)
";AN$
290 IF AN$="" THEN 280
300 S=ASC(UPPER$(AN$))-64:IF S<0 OR S>15
 THEN 280
310 Q(Q,N,0)=S
320 INPUT #3,"Rendezvous (@ or A to C)";
AR$
330 IF AR$="" THEN 320
340 R=ASC(UPPER$(AR$))-64:IF R<0 OR R>3
THEN 320
350 Q(Q,N,1)=R
360 LOCATE #4,1+2*(N-1),1+2*(Q-1):PRINT
#4,CHR$(Q(Q,N,0)+64);
370 LOCATE #4, 1+2*(N-1),2*Q:PRINT #4,R$
;:IF Q(Q,N,1)>0 THEN PRINT #4,CHR$(Q(Q,N
,1)+64);:ELSE PRINT #4," ";
380 PRINT #4,R$;:NEXT
390 NEXT
400 GOTO 80
410 GOSUB 1020
420 A$="Y":WHILE A$="Y"
430 CLS #3:INPUT #3,"Sound letter (A to
O)";AS$
440 IF AS$="" THEN 430
450 S=ASC(UPPER$(AS$))-64:IF S<1 OR S>15
```

```
 THEN 430
460 AM$="Tone period (0 to 4095)":LL=0:U
L=4095:TB=2:GOSUB 970:S(S,2)=M
470 AM$="Duration (-32767 to 32767)":LL=
-32767:UL=32767:TB=3:GOSUB 970:S(S,3)=M

480 AM$="Initial volume (0-15)":LL=0:UL=
15:TB=4:GOSUB 970:S(S,4)=M
490 INPUT #3,"Envelopes or noise (Y/N)";
AN$
500 IF UPPER$(AN$)="N" THEN 550
510 CLS #3
520 AM$="Volume envelope (0 to 15)":LL=0
:UL=15:TB=1:GOSUB 970:S(S,5)=M
530 AM$="Tone envelope (0 to 15)":LL=0:U
L=15:TB=2:GOSUB 970:S(S,6)=M
540 AM$="Noise (0 to 15)":LL=0:UL=15:TB=
3:GOSUB 970:S(S,7)=M
550 LOCATE #1, 1,S+2:GOSUB 1060
560 INPUT #3,"Another sound (Y/N)";A$
570 A$=UPPER$(A$)
580 WEND:GOTO 80
590 VT=15:GOSUB 870
600 A$="Y":WHILE A$="Y"
610 CLS #3:AM$="Envelope number (0 to 15
)":LL=0:UL=15:TB=3:GOSUB 970:E=M
620 IF M=0 THEN 80
630 FOR SE=0 TO 4:CLS #3:PRINT #3,"Secti
on "SE+1
640 AM$="Step count (0 to 127)":LL=0:UL=
127:TB=2:GOSUB 970:E(E,16+3*SE)=M
650 AM$="Step size (-128 to 127)":LL=-12
8:UL=127:TB=3:GOSUB 970:E(E,17+3*SE)=M

660 AM$="Pause time (0 to 255)":LL=0:UL=
255:TB=4:GOSUB 970:E(E,18+3*SE)=M
670 ENV E,E(E,16),E(E,17),E(E,18),E(E,19
),E(E,20),E(E,21),E(E,22),E(E,23),E(E,24
),E(E,25),E(E,26),E(E,27),E(E,28),E(E,29
),E(E,30)
680 NEXT:CLS #3
690 LOCATE #2, 1,E+2:GOSUB 940
700 INPUT #3,"Another envelope (Y/N)";A$

710 A$=UPPER$(A$)
```

```
720 WEND:GOTO 80
730 VT=0:GOSUB 870
740 A$="Y":WHILE A$="Y"
750 CLS #3:AM$="Envelope number (-15 to
15)":LL=-15:UL=15:TB=3:GOSUB 970:E=ABS(M
):E(E,0)=SGN(M)
760 IF M=0 THEN 80
770 FOR SE=0 TO 4:CLS #3:PRINT #3,"Secti
on "SE+1
780 AM$="Step count (0 to 127)":LL=0:UL=
127:TB=2:GOSUB 970:E(E,1+3*SE)=M
790 AM$="Step size (-128 to 127)":LL=-12
8:UL=127:TB=3:GOSUB 970:E(E,2+3*SE)=M
800 AM$="Pause time (0 to 255)":LL=0:UL=
255:TB=4:GOSUB 970:E(E,3+3*SE)=M
810 ENT E(E,0)*E,E(E,1),E(E,2),E(E,3),E(
E,4),E(E,5),E(E,6),E(E,7),E(E,8),E(E,9),
E(E,10),E(E,11),E(E,12),E(E,13),E(E,14),
E(E,15)

820 NEXT:CLS #3
830 LOCATE #2, 1,E+2:GOSUB 940
840 INPUT #3,"Another envelope (Y/N)";A$

850 A$=UPPER$(A$)
860 WEND:GOTO 80
870 CLS #2
880 IF VT=0 THEN 900
890 PRINT #2,"VOLUME ENVELOPES":PRINT #2
,"No  P1  Q1 R1  P2  Q2 R2  P3  Q3 R3  P
4  Q4 R4  P5  Q5 R5";:GOTO 910
900 PRINT #2,"TONE ENVELOPES":PRINT #2,"
No  T1  V1 W1  T2  V2 W2  T3  V3 W3  T4
 V4 W4  T5  V5 W5";
910 FOR E=1 TO 15
920 GOSUB 940
930 NEXT:RETURN
940 PRINT #2, USING Z2$;E;:FOR P=0 TO 4
950 PRINT #2," ";R$;:PRINT #2, USING Z3$
;E(E,1+P*3+VT);:PRINT #2,R$;:PRINT #2, U
SING Z4$;E(E,2+P*3+VT);:PRINT #2,R$;:PRI
NT #2, USING
 Z3$;E(E,3+P*3+VT);:PRINT #2,R$;
960 NEXT:RETURN
970 M=-32768:WHILE M<LL OR M>UL
```

```
980 LOCATE #3,1,TB:PRINT #3,AM$;:INPUT #
3," ",A$
990 IF A$="" THEN 980
1000 M=VAL(A$)
1010 WEND:RETURN
1020 CLS #1:PRINT #1,"SOUNDS":PRINT #1,"
 Ch Tone  Dur  V v t N";
1030 FOR S=1 TO 15
1040 GOSUB 1060
1050 NEXT:RETURN
1060 PRINT #1,CHR$(64+S);R$;:PRINT #1, U
SING Z3$;S(S,1);:PRINT #1,R$;:PRINT #1,
USING Z4$;S(S,2);:PRINT #1,R$;:PRINT #1,
 USING Z6$;S
(S,3);:PRINT #1,R$;
1070 FOR SP=0 TO 3
1080 PRINT #1, USING Z2$;S(S,4+SP);:PRIN
T #1,R$;
1090 NEXT:RETURN
1100 CLS #4
1110 FOR Q=1 TO 3
1120 FOR N=1 TO 5:PRINT #4,CHR$(Q(Q,N,0)
+64);" ";:NEXT
1130 PRINT #4,R$;:FOR N=1 TO 5:IF Q(Q,N,
1)>0 THEN PRINT #4,CHR$(Q(Q,N,1)+64);:EL
SE PRINT #4," ";
1140 PRINT #4," ";:NEXT

1150 PRINT #4,R$;:NEXT
1160 RETURN
```

14

What Next?

When you have a decision to take, be it in your private life, your career or business, it usually helps to set down on paper all the possible events that could occur, all possible actions that you could take and what their consequences might be. Edward de Bono recommends an approach of this sort to most of life's problems. This program takes you one stage further. It helps you find your way through the confusing and often contradictory mass of facts and assumptions and arrive at what is likely to be the best course of action.

The program is based on techniques of operational analysis that are applicable to decision-taking. The program is a relatively simple one and therefore has some limitations, as will be explained later. It is not infallible but, in spite of this, can give you useful pointers as to what your next step should be.

It might also be said that, by the time you have sat down and have thought about and listed the possibilities and the courses of action open to you, you will have arrived at a deeper understanding of what confronts you. You will be able to proceed to the best decision without having to use the program. If this is the case, the program will have achieved its stated objective!

What the program does

The program supposes that you are in a situation in which one or more of a number of different events (at least two, not more than ten) may occur and that you have a number of courses of action (at least two, not more than ten) available to you. The program will tell you which one or more of your courses of action will give you the best reward in the long run. Note the words 'in the long run'. The program assumes that this is not just a single once-and-for-all decision, but a decision involving a continuing policy. No program could be expected to tell you exactly what will happen on a single occasion. The program also assumes that your opponent also has access to the results of running the program and will always act so as to make things as bad as possible for you.

For the first example we will consider the decisions required in playing a simple game. We have chosen a game for this introductory example because a game has rules. The events that may occur and the actions that we may take are both governed by these rules, so that we can have a precise idea of what will happen. The rules also tell us how to work out the scores of the winner and the loser. The game is the traditional one called Two-Finger Morra, a game for two players. On the word 'go', both players raise their hands, showing either one or two fingers. At the same time, each player says 'one' or 'two', trying to guess how many fingers the other player will show. If both players guess correctly, or both guess incorrectly, nothing further happens in that round. But if just one player guesses correctly, this player wins. The loser pays the winner an amount equal to the total number of fingers shown by both.

Suppose that you are player A and your opponent is player B. What B does is out of your control; it is referred to as an *event*. The four possible events are:

B shows 1 finger and guesses 1.
B shows 1 finger and guesses 2.
B shows 2 fingers and guesses 1.
B shows 2 fingers and guesses 2.

These are all the events the rules allow. Since the same rules apply to both players, you are able to do the same four things. When you do them they are referred as *actions*. The actions are under your control. In the table below we set out all possible events and actions:

	EVENTS (by B)			
	Show 1; guess 1	Show 1; guess 2	Show 2; guess 1	Show 2; guess 2
ACTIONS (by you, A)				
Show 1, guess 1	0	2	−3	0
Show 1; guess 2	−2	0	0	3
Show 2; guess 1	3	0	0	−4
Show 2; guess 2	0	−3	4	0

The cells of the table show what amount B is to pay you at the end of each round. When you both guess correctly or both guess incorrectly, the amount is zero. A positive amount shows that you alone guessed correctly, and won from B. A negative amount shows that B alone guessed correctly, and you lost to B. The amount in each cell is called the payoff, and this is a payoff table.

When we use the program (as explained later) we key in the payoff table. The computer then finds out which action or which combination (or mix) of actions you should use to obtain the maximum payoff. In doing this, the program assumes that B also knows the payoffs and will always act so as to make your payoff as small as possible. B is always out to beat you.

When the table above is keyed in, the program tells us that the best mix of actions for you is to show 1 finger and guess 2 on 57% of occasions and to show 2 fingers and guess 1 on the remaining 42% of occasions (the percentages do not necessarily add up exactly to 100%). How the computer arrives at this advice cannot be explained in the limited space available here. What the advice means is that, over a game of several rounds, you should always play in one of the two ways suggested. You should adopt one action on 57% of rounds, and the other on the 42% of rounds, but do this in an unpredictable way, of course, so that B cannot tell what you intend to do next.

The program also works out the expected average payoff for each round, supposing that you follow its advice. Disappointingly, the payoff is zero. In a game of many rounds you are likely to finish with as much money in your pocket as you started with. If you have more (or less), it will be simply good (or bad) luck! This is to be expected, for the rules do not favour one player or the other and it is just a guessing game in which both players have equal chances of success.

Now for an example in which the 'rules' are not so clear-cut, and in which they are *not* exactly the same for both 'players'. Suppose you are running a Social Club which meets every Saturday night. Your club has a barbecue, disco equipment, a video recorder and a hall with badminton courts. There is one other club (call it Club B) in the same town which has much better disco facilities than your club. It also has a hall with a good selection of gymnastic apparatus and a fully equipped stage. Your club puts on a program during the year, in which each Saturday night is devoted to one type of entertainment, either a barbecue supper, a disco, a video show, or badminton. The other club also has a Saturday night program, of either disco, keep fit classes or a drama group. Your aim is to increase the membership of your club, and we will assume that if your club gains members, Club B loses them, or vice versa. We will also assume that most people will join one club or the other and, having joined, will not readily switch from one to the other and back again – at least, not week-by-week. In other words, you have to organise and advertise a program which will keep your own members happy and induce members of Club B to leave that club and join yours.

To help you plan your program, you draw up a table showing the *events* (organised by Club B) and the *actions* (organised by you). The payoffs in the table are your estimates of how many members you would expect to gain when two kinds of entertainment are presented on the same Saturday evening.

	EVENTS (Club B)		
	super disco	keep fit class	drama group
ACTIONS (your club)			
barbecue supper	30	0	−15
disco	−40	10	50
video show	20	−10	40
badminton	−50	25	10

The figure −40, for example, is your estimate of how many members you would lose when your disco was in competition with the superior disco of Club B. The 40 shows how many members you expect to gain if you put on a video show on the same evening as the drama group. These are your estimates, and the reliability of the advice given by the program can be no greater than the reliability of your estimates.

When the payoff table above is fed into the computer the advice it gives is to mix your actions in the following proportions:

Barbecue supper	48%
Disco	0%
Video show	23%
Badminton	28%

It seems that you should abandon the disco, and plan your program to include the other three entertainments in the proportions indicated. This will ensure that you obtain the maximum number of memberships from former Club B members. The expected payoff is given as 5, meaning that by adopting this advice you can expect to increase your membership by 5. Note that the program indicates that you have a definite chance of increasing your membership, presumably because of the greater popularity of the entertainment you offer.

It is possible, though it is not strictly valid, to use the program when your opponent is not another person or organisation but is 'blind fate'. A good example of 'fate' is the weather. Suppose you are planning how to spend your weekends, in order to give yourself the maximum happiness. Incidentally payoffs can be money, members, happiness or anything else that you feel that you can express with a numerical value. The events of this example are the weather, which may be warm and sunny, cold and windy, or rainy. Your possible actions are to go to the seaside, to visit a stately home, to visit a leisure park, to go to a cinema, or to stay at home. Here is your payoff table, in which positive payoffs indicate happiness, while negative ones indicate unhappiness.

	EVENTS (fate)		
	Warm and sunny	Cold and windy	Rainy
ACTIONS (you)			
Stay at home	−30	20	10
Seaside	100	−50	−80
Stately home	20	10	5
Leisure park	−30	20	10
Cinema	−50	40	50

It is clear from this table that you love the seaside on a sunny day, but find it awful in bad weather. The cinema is your first choice for a wet day but, if it is sunny, you do not enjoy it; you would rather have gone to the sea. So what mix of weekend activity is likely to give you the most happiness in the long run? The program gives this advice:

Stately home	86%
Cinema	13%

It gives zero for all others. This is not surprising for the leisure park and for staying at home. You have not rated these high for happiness on average. Although you have rated the seaside very high on a sunny day, you have rated it very low in bad weather. This brings out a point concerning this program when used with 'fate' as 'player B'. The program assumes that player B always does its best to make things as bad as possible for you. Many people feel that the weather (or any other aspect of fate) is indeed always against them. If this is how you feel, then the program reflects your expectations fairly accurately. Forget about going to the seaside for, if you do, the weather is certain to be awful!

The payoff of this table is 11, so it seems that you are of a relatively happy disposition and, provided that you go to the stately homes six to seven times as often as you go to the cinema you will be reasonably happy most weekends. You seem to enjoy stately homes moderately in any weather, so the computer advises you to spend most of your weekends visiting these. However, after visiting all the stately homes in the neighbourhood, boredom might offset your happiness. If you then decide to give them 'a miss' for a while, leave out 'stately home'. You are then recommended to try the seaside (35%) and the cinema (64%). Payoff is 4. This illustrates the point that as time passes you should reassess the payoffs and alter your strategy accordingly.

Using the program

Before you use the program, write out a payoff table like those in the examples given above. There must be between 2 and 10 events (columns) and between 2 and 10 actions (rows). There need not be the same number of each. In any table, payoffs must all represent the same thing, such as money, happiness, or time.

When you run the program, the computer first asks how many events there are and how many actions you have available. Next the computer asks you to key in the payoffs. Key these in column by column. If you make a mistake, press ESCAPE twice, rerun the program and begin again. As soon as the last payoff has been entered, the screen clears. The message 'Calculating' appears. A few seconds later the analysis is complete. You will see that either one action or a mixture of actions is recommended. In the latter case, you are told what the 'mix' should be. Actions which are not to be recommended are often eliminated early in the calculation. If this happens, the actions concerned are not mentioned in the display. Actions eliminated at a later stage in the calculation are listed in the display, but with the percentage '0%'. If you are told to mix two or more actions, on some occasions you should take one action and on other occasions one of the others. Over a series of occasions, the actions should be taken at random (do not let B know in advance) with the frequencies indicated by the percentages. The result of this strategy is to give you a payoff close to the value displayed.

Keying in

Although it takes the micro only a few seconds to produce the advice, the calculation is an extremely elaborate one which could take you an hour to perform on paper. Type the program accurately, paying special attention to the N's, M's, MM's, MY's and so on.

Program design

20–30 Initialisation.
40–130 Inputting payoff table.
140–690 Analysing the table to find if there are single actions to be recommended; deleting actions which are certainly not worth taking under any circumstances.
700–720 Preparing a table for simplex analysis.
730–910 Simplex analysis to determine the best mix of actions.
920–990 Display of mix of actions.
1000–1020 Display of single best action.

The program

```
10 REM ** WHAT NEXT **
20 MODE 1
30 DIM T(22,11),MR(10),MC(10),D(10,10),D
R(10),E(10,10),ER(10)
40 X=0:CLS:INPUT "How many events";M$
50 M=VAL(M$):IF M<2 OR M>10 THEN 40
60 INPUT "How many actions";N$
70 N=VAL(N$):IF N<2 OR N>10 THEN 60
80 CLS:PRINT "Enter payoffs"
90 FOR J=1 TO M
100 PRINT "EVENT";J
110 FOR K=1 TO N
120 PRINT "    Action"K;:INPUT" ";T(J,K)
130 NEXT:NEXT
140 CLS:PRINT "Calculating"
150 FOR K=1 TO N:MR(K)=T(I,K)
160 FOR L=2 TO M
170 IF T(L,K)<MR(K) THEN MR(K)=T(L,K)
180 NEXT:NEXT
```

```
190 MA=MR(1):BA=1
200 FOR K=2 TO N
210 IF MR(K)>MA THEN MA=MR(K):BA=K
220 NEXT
230 FOR J=1 TO M
240 MC(J)=T(J,1)
250 FOR L=2 TO N
260 IF T(J,L)>MC(J) THEN MC(J)=T(J,L)
270 NEXT:NEXT
280 MY=MC(1)
290 FOR J=2 TO M
300 IF MC(J)<MY THEN MY=MC(J)
310 NEXT
320 IF MA=MY THEN 1000
330 IF N<3 THEN 490
340 FOR K=1 TO N
350 FOR L=1 TO N
360 FOR J=1 TO M
370 IF T(J,K)>=T(J,L) THEN 390
380 D(K,L)=1
390 NEXT:NEXT:NEXT
400 FOR K=1 TO N
410 FOR L=1 TO N
420 IF D(K,L)=0 AND K<>L THEN DR(L)=1
430 NEXT:NEXT
440 K=1:L=1
450 IF DR(K)=0 THEN DR(L)=K:L=L+1
460 K=K+1:IF K<N+1 THEN 450
470 N=N+L-K:FOR K=1 TO N:FOR J=1 TO M:T(
J,K)=T(J,DR(K)):NEXT:NEXT
480 IF N=1 THEN BA=DR(1):GOTO 1000
490 IF M<3 THEN 610
500 FOR J=1 TO M:FOR L=1 TO M:FOR K=1 TO
 N
510 IF T(J,K)<=T(L,K) THEN 530
520 E(J,L)=1
530 NEXT:NEXT:NEXT
540 FOR J=1 TO M
550 FOR L=1 TO M
560 IF E(J,L)=0 AND J<>L THEN ER(L)=1
570 NEXT:NEXT
580 J=1:L=1
590 IF ER(J)=0 THEN ER(L)=J:L=L+1
600 J=J+1:IF J<M+1 THEN 590
610 M=M+L-J:FOR J=1 TO M:FOR K=1 TO N:T(
J,K)=T(ER(J),K):NEXT:NEXT
620 FOR J=M+1 TO M+N:FOR K=1 TO N:T(J,K)
=0:NEXT:NEXT
630 MM=T(1,1):FOR J=1 TO M:FOR K=1 TO N:
IF T(J,K)<MM THEN MM=T(J,K)
640 NEXT:NEXT
650 IF MM>0 THEN MM=0:GOTO 700
660 MM=1-MM
670 FOR J=1 TO M:FOR K=1 TO N
680 T(J,K)=T(J,K)+MM
690 NEXT:NEXT
700 T(0,0)=1
710 FOR J=1 TO M:T(J,0)=-1:NEXT
720 FOR K=1 TO N:T(M+K,K)=1:T(M+N+1,K)=1
:NEXT
730 L=0:FOR J=1 TO M+N
740 IF T(J,0)<L THEN L=T(J,0):C=J
750 NEXT
760 IF L=0 THEN GOTO 920
770 MR=1E+10:FOR K=1 TO N
780 X=T(C,K):B=T(M+N+1,K)
790 IF X<=0 THEN GOTO 810
800 TR=B/X:IF TR<MR THEN MR=TR:R=K
810 NEXT
820 P=T(C,R):FOR J=1 TO M+N+1
830 T(J,R)=T(J,R)/P
840 NEXT
850 FOR K=0 TO N
860 F=T(C,K)/T(C,R):IF K=R THEN 900
870 FOR J=1 TO M+N+1
880 T(J,K)=T(J,K)-F*T(J,R)
890 NEXT
900 NEXT
910 GOTO 730
920 CLS:LOCATE 5,5:PRINT "The best blend
 of actions is:"
930 FOR K=1 TO N
940 LOCATE 5,(5+K*2):PRINT "Action "DR(K
)":   "INT(T(M+K,0)/T(M+N+1,0)*100)"%"

950 NEXT
960 LOCATE 5,15:PRINT "Expected payoff i
s: "1/T(M+N+1,0)-MM
970 LOCATE 3,17:PRINT "<Space bar for an
other assessment>"
```

```
980 A$=INKEY$:IF A$<>" " GOTO 980
990 RUN
1000 CLS:LOCATE 10,5:PRINT "The best act
ion is: "BA
1010 LOCATE 10,7:PRINT "Expected payoff
is: "MA
1020 GOTO 970
```

GET THE AMSTRAD WORKING FOR YOU!

Here are fourteen useful, highly practical programs for your Amstrad.

Programs include:

- **Accounts – an accounting program for the home or small business.**
- **Stock Take – to keep track of your supplies whether in the kitchen, workshop or office.**
- **Cashflow – a spreadsheet program.**
- **Pools Punter – record past results and use them to predict the next match.**
- **Supersound – makes the most effective use of the Amstrad's excellent but complex sound facilities.**
- **Index – invaluable for all kinds of alphabetical indexing – books, catalogues, collections, etc.**
- **Trend Finder – are things getting better or worse? This program helps you find out.**

Full instructions and suggested applications are provided, with all you need to know to adapt the programs to suit you best.

The Authors

Owen Bishop and Audrey Bishop are freelance technical writers and programmers. Between them they have written over forty books, including many on popular computing. Owen Bishop is a well-known and regular contributor to computing journals.

Other books for Amstrad users

AMSTRAD COMPUTING
Ian Sinclair
0 00 383120 5

SENSATIONAL GAMES FOR THE AMSTRAD CPC464
Jim Gregory
0 00 383121 3

40 EDUCATIONAL GAMES FOR THE AMSTRAD CPC464
Vince Apps
0 00 383119 1

ADVENTURE GAMES FOR THE AMSTRAD CPC464
A. J. Bradbury
0 00 383078 0

Front cover illustration by Tony Roberts

COLLINS
Printed in Great Britain
0 00 383082 9

£6.95 net

BISHOP AND BISHOP

PRACTICAL PROGRAMS FOR THE AMSTRAD CPC464

COLLINS

AMSTRAD CPC

MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA

https://acpc.me/